

Designing the Interactive Architecture Space:  
A User-centered Framework for Designing Interactive Sensing Environments

Runchang Kang  
Carnegie Mellon University

**CARNEGIE MELLON UNIVERSITY**

**School of Architecture**  
College of Fine Arts

**Thesis**

Submitted in Partial Fulfillment of the requirements for the degree of

**Master of Science in Computational Design**

**TITLE:**

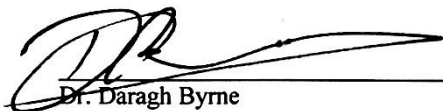
**Designing the Interactive Architecture Space:**

**A User-centered Framework for Designing Interactive Sensing Environments**

**AUTHOR:**

**Runchang Kang**

**ACCEPTED BY ADVISORY COMMITTEE:**



Dr. Daragh Byrne

Principle Advisor



Dr. Daniel Cardoso Llach

Committee Chair

**MAY 15<sup>th</sup> 2019**

DATE

**MAY 15, 2019**

DATE

### Acknowledgments

I am sincerely thankful to the following individuals for their encouragement, support and inspiration.

Prof. Daragh Byrne for his guidance during this research. For his profound knowledge in the realms of interactive environment, Internet-of-things, and human-computer interaction; for providing suggestions from solid technical perspective; and for the great number of hours of advice.

Prof. Daniel Cardoso Llach for cultivating me to be a researcher, for introducing me to the realm of Computational Design, for teaching me how to conduct research, and countless of great insights and suggestions to my research.

The members of the Code Lab, Wei-Wei, Oz, Jinmo, Humphrey, Zheng, Zhuoni for many hours of great discussion.

The researchers from HCII, Gierad, Anthony, Sai, Anhong for the brainstorming and technical support.

My family and friends.

## Abstract

Due to rapid developments in computation, cybernetics, artificial intelligence, and engineering, and the architectural environment, the line between the physical and the digital world is increasingly blurred. Our physical spaces can now become more intelligent: receptive, perceptive, reactive, interactive, and even proactive in its responses to human occupants. In practice, however, achieving these reactive spaces is challenging. They require complex and integrated systems for sensing occupants' status, activities, interactions to provide contextual information as inputs. This is further compounded by the diversity of appliances and applications that could be integrated sensing system to create outputs and effects in the environment. The technical complexity involved makes the development of such environments expensive in time and effort, and challenged by extensibility and expandability. This also creates significant barriers for non-technical experts, i.e. designers and architects, to participate in the preparation of interactive spaces.

In this thesis, I present an indoor context sensing system that can gather, analyze and fuse the occupants' movement, gestures and voice data. It also provides easy-to-use templates and rules for designers and developers to quickly integrate smart appliances and apply rules for easily configuration of these appliances. Finally, I demonstrate the potential of the system as a toolkit for rapid prototyping of interactive environments in two major domains - smart home context and new-media installations. The system, including software and hardware components, is packaged and well-documented, serving as an off-the-shelf, open-source project for developing interactive, intelligent environments.

## Table of Contents

|                                                            |    |
|------------------------------------------------------------|----|
| Acknowledgments.....                                       | 3  |
| Abstract.....                                              | 4  |
| CHAPTER 1 INTRODUCTION .....                               | 7  |
| 1.1 Interactive Architecture and Reactive Space .....      | 7  |
| 1.2 That Sounds Fascinating, But What is the Reality?..... | 10 |
| 1.3 Computer-Aided Interactive Architecture Design .....   | 12 |
| CHAPTER 2 BACKGROUND .....                                 | 15 |
| 2.1 Distributed Cognition and Embodied Interaction .....   | 15 |
| 2.2 Internet of Things.....                                | 16 |
| 2.3 Interactive Architecture.....                          | 20 |
| 2.4 Multimodal Interactions.....                           | 24 |
| 2.5 End-user Development.....                              | 27 |
| CHAPTER 3 RESEARCH DESIGN AND METHODOLOGY .....            | 31 |
| 3.1 Preliminary Formative Study .....                      | 31 |
| 3.2 System Design and Implementation .....                 | 34 |
| 3.2.1 Centric Backend System .....                         | 35 |
| 3.2.2 End-user Hardware Wristband and Appliances .....     | 44 |
| 3.2.3 End-user Template and Configuration File.....        | 49 |
| 3.3 Configuration Instruction.....                         | 54 |

|                                      |    |
|--------------------------------------|----|
| CHAPTER 4 EXAMPLE APPLICATION .....  | 56 |
| 4.1 Example Application.....         | 56 |
| 4.1.1 Spatial Piano .....            | 56 |
| 4.1.2 Voice Neon.....                | 57 |
| 4.1.3 Smarter Home .....             | 58 |
| CHAPTER 5 HEURISTIC USER STUDY ..... | 60 |
| CONCLUSION and FUTUREWORKS .....     | 63 |
| References.....                      | 65 |
| Appendix.....                        | 70 |
| Appliance Template Java Code:.....   | 70 |
| Rule Template Java Code:.....        | 74 |

## CHAPTER 1 INTRODUCTION

### 1.1 Interactive Architecture and Reactive Space

Experiencing an architectural space is like having a dialog between the occupants and space. The dialog can be more expressive and interactive with the help of technology. Michael Fox's definition is "*Interactive architecture is built upon the convergence of embedded computation and a physical counterpart that satisfies adaptation within the framework of interaction. It encompasses both buildings and environments that have been designed to respond, adapt, change, and come to life.*"<sup>1</sup> The idea of architecture breaking out of the static, cold, and lifeless stereotypes and becoming an active participant with the capacity to perceive, react to, and connect with the occupants and with the natural world, is catching more and more attention. "*Our architectural surroundings have become so inextricably tied to technological trends that the two ultimately and simultaneously respond to and define each other.*"<sup>2</sup> Indeed, the architectural environment as the most dominant context within our daily life is also the most suitable testbed for eliciting new technology and novel ideas. Embedded devices, sensors, actuators, cloud techniques, web services have already gradually infiltrated the infrastructure of the built space in order to make observations, process information, offer recommendations, and provide services in order to connect us to the rest of the world and create seamlessly integrated environments. Mark Weiser named these scenarios as Ubiquitous Computing<sup>3</sup>: "*The idea of integrating computers seamlessly into the world at large runs counter to a number of present-day trends.*"<sup>4</sup> which serves as the foundation for countless exciting computational domains.

---

<sup>1</sup> Fox, Michael. *Interactive Architecture Adaptive World*. New York: Princeton Architectural Press, 2016.

<sup>2</sup> Fox, Michael. *Interactive Architecture Adaptive World*. New York: Princeton Architectural Press, 2016.

<sup>3</sup> Weiser, Mark. "The Computer for the 21st Century." *ACM SIGMOBILE Mobile Computing and Communications Review* 3, no. 3 (07, 1999): 3-11. doi:10.1145/329124.329126.

<sup>4</sup> Weiser, Mark. "The Computer for the 21st Century." *ACM SIGMOBILE Mobile Computing and Communications Review* 3, no. 3 (07, 1999): 3-11. doi:10.1145/329124.329126.

Early 1960s research in cybernetics served as the theoretical groundwork for interactive architecture, basically establishing a model by which architects interpreted spaces and users as complete feedback systems. Later, John Frazer, as one of the pioneers of the use of computers in architecture, envisioned the future of architecture is to construct organic buildings.<sup>5</sup>

Thermodynamically open systems which are more environmentally aware and sustainable physically, sociologically and economically<sup>6</sup>. In the meantime, fields of embedded computation, ambient intelligence, and human-computer interaction were moving forward rapidly within the realm of computer science. This interdisciplinary research tried to bring computational technologies into the physical world in order to enhance people's quality of life, communication efficiency, and improve productivity. Meanwhile, established market-driven interests that played a major role in computationally enhanced environments through the development of numerous market-driven products and systems that directly involved users in the real world.<sup>7</sup> The "Smart Home" and "Smart Office" concept has gradually been accepted by the public. For example, consumer electronic companies created appliances for home automation which allows the user to connect, control, and even predict services. At the same time, the idea of the interactive and connected environment also drew the attention of many young designers. Countless of interactive installations and responsive space projects were developed, built upon the key concept of the Internet of Things, which defined the technology context for the interactive design and extended to affect essentially everything, from digital representation to functional installations to the reactive environment and even to the smart connected city.

---

<sup>5</sup> "John Frazer." John Frazer Researcher. <http://www.johnfrazer.com/research>

<sup>6</sup> Frazer, John. *An Evolutionary Architecture*. London: Architectural Association, 1995.

<sup>7</sup> Fox, Michael. *Interactive Architecture Adaptive World*. New York: Princeton Architectural Press, 2016.

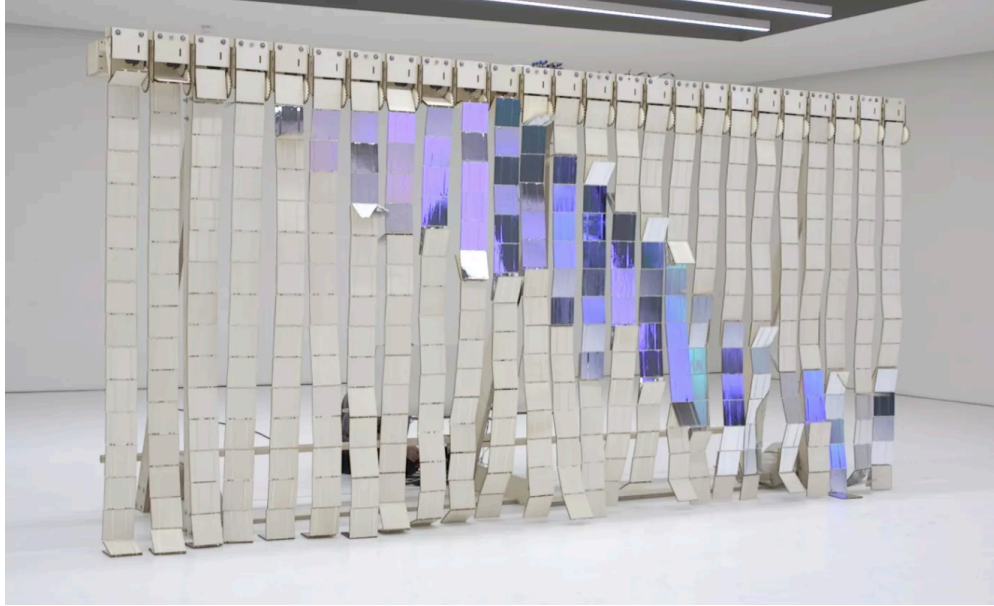


*Figure 1: Analog Future, Parker Heyl, Interactive Architecture Lab, UCL, an interactive architecture project combines the experience of sound, light and space*

Looking into the built space, individual occupants are within a constant dialogue with the surrounding space, and the space, as one of the most important considerations through the whole lifecycle of architectural design, is extended to one more dimension, represented by the reactions, the responses, and the transformation from the devices that embedded in the infrastructure of the environment. Scott Fisher, the founding chair of the Media Arts + Practice of the School of Cinematic Arts at the University of Southern California, mentioned: *“The growing number of ubiquitous and embedded computing technologies introduces a new paradigm for how we interact with the built environment, while mobile and pervasive devices offer new possibilities for sensing and communicating with buildings and objects in the physical world. These technologies are used not only for collecting and providing data but also as a way to animate and collectively augment the world around us.”*<sup>8</sup> These technologies are not only the media or

---

<sup>8</sup> Scott S. Fisher, Jennifer Stein, *Interactive Architecture: Connecting and Animating the Built Environment with the Internet of Things*, Internet of Things Conference Workshop, Tokyo, 2010

carriers of the data and information but also substantially augmenting the world around us.

Following this idea, many people were trying to understand how these networked spaces and objects might become playfully responsive characters that engage people with the built environment in new ways.

## 1.2 That Sounds Fascinating, But What is the Reality?

Although designing interactive architecture and creating responsive gadgets distributed in space has become a popular approach for designers to express their thinking on the human-space relationship and humanistic care in providing convenience and comfort for the occupant. Setting up such a system still requires great effort. Most of the existing reactive environment projects are bound to a very specific context, designed and deployed to augment the environment in a monotonous way. Scalability and extensibility become the most essential problem of those systems. For example, the SIX-FORTY BY FOUR-EIGHTY project by Zigelbaum and Coelho<sup>9</sup> has to use the UART ports for transmitting the data between module, which will be hard for building up an interactive project upon hundreds of gadgets. On the other hand, the interaction between the occupants and the system still remains unclear for many existing sensing projects, which requires the occupant great effort to understand and communicate with the system. Berado et al. presented an experimental design framework project which concentrates to create personalized indoor microclimates through the feedback loops between responsive thermal system and the occupants' biometrics. (Figure 2) "... the goal of this study is to characterize an

---

<sup>9</sup> Zigelbaum, Jamie, and Marcelo Coelho. "Six-Forty by Four-Eighty." *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '11*, 2011, doi:10.1145/1979742.1979564.

experimental framework that integrates responsive thermal systems with occupants' direct and indirect experience, which includes stress response and biometric data.” A physical installation was built to test the model. They created a suspended modular radiant ceiling which would change the temperature and radiation for local microclimate zones in reacting to the biometric data getting from the wearable biometric sensors and the thermal data from an IR camera. The participants were required to sitting in the 8'×10' footprint area to simulate the living condition and they conducted the research to prove their research concept. However, although fusing the thermal and biometric data for improving the living experience is an interesting research orientation, their system wasn't designed wisely in response to how occupants experiencing and living in a built environment. The movement of the user, which is very essential for the built environment, wasn't considered in this project.



*Figure 2: Berardo Matalucci, Kenton Phillips, Alicia A Wal, Anna Dyson and Joshua Draper, An experimental design framework for the personalization of indoor microclimates through feedback loops between responsive thermal systems and occupant biometrics.*

Moreover, the application and appliance compatibility are not generic enough, many of the applications were hardcoded to specifically work for that system. For example, due to the different protocols implementation and various of system interfaces, it will be hard for an

appliance to be re-use in another interactive project's system. If the appliance can have a generic template to follow in order to make it compatible across different system, it will increase the scalability and extensibility of the system. Because there is no a user-friendly ecosystem for designing and deploying reactive appliances in sensing space, each company has their own version of application templates, which coupling the design of applications to the system itself. For example, the Philip Hue can only be controlled by the mobile app provided by Philip in their early released, and it took great engineering effort for them to transfer the system compatible with the other intelligence voice assistants like Alexa. On the other hand, the disunity of appliance templates creates barriers to end-user development, which makes the sensing system can only be used in a very particular way, following the system designer's original purpose and hard to configure the appliance differently according to user's preference. For example, the end-user has limited access to the behavior and functionality of the commercial smart device, which required them to re-flash the firmware if they need to create their own application based on the commercial products.

### 1.3 Computer-Aided Interactive Architecture Design

As I mentioned in the previous section, although creating interactive space and the responsive environment has drawn many designers and researchers' interest, most of the existing infrastructure systems are not designed and optimized for building the interactive architecture. Starting from this perspective, I raised the main research question: **How to enable rapid prototyping design processes for creating extensible and efficient interactive spaces which also provides intuitive interaction and closer relationships between the occupant, device, and the environment?** Designing the reactive appliances which are distributed in the intelligent

environment and configuring the behavioral reaction of them by responding to the occupants' status, movement and activities would be the keys for the research.

For getting a more feasible research sequence, I broke down the main research question to three sub questions.

1). First, *discovering the key capabilities that the proposed smart interactive system* should have is essential. This will help us to understand the question of “*how*” an end-user will interact with the system. By referencing the precedents, finding expert opinions, and conducting a formative user study, I was able to conclude that end-user tends to interact with the environment in their familiar way. To be more specific, most of the participants were willing to interact with the system in a similar way that people interact with each other. According to that, understanding the occupants' gestures, voices, activities, locations, movements would be crucial for this intelligent system.

2). After probing the key capabilities of the system, the following sub-question arises: *how should the system be implemented such that it those key capabilities but it also a) maintains high scalability and extensibility for future development and b) is readily adaptable to a range of interactive/project contexts.* Through a series of iteration and attempts, I integrated a real-time tracking sensor for localizing the end-user, a motion sensor for tracking the hand gestures, and Natural Language Processing techniques for understanding the verbal commands. Each component works independently but also collaboratively because of a lightweight server was used as the data protocol, which makes each functional component only needs to focus on itself and publishes individual data to the server.

3) After constructing an integrated sensing system informed by the findings from the first and second sub-questions, the last but the most important question is raised as: *how does the*

*system reduce the time and difficulty of, as well as increase the convenience of, designing reactive appliances for end-users?* Previous systems such as require end-user great effort to configure the appliance, program the behavior, and add extra functionalities. Building a pipeline which would tackle those problems individually is the imperative part of the research. By comparing several existing solutions and balancing the tradeoffs between each of them, my system exposes extensibilities in two-fold by providing user with Appliance Template and Rule Template, one of which allows the designer to easily configure the functionalities of the appliance that he/she wants to register into the system, the other provides them to define trigger rules which would be automatically ensure the execution in runtime by the system.

Ultimately, this research also delivers a packaged version of the system code and the implementation details of the software and hardware design to serve as an open-source project for designers to create their own the interactive environment design projects, as well as elicit further improvement and creativity.

## CHAPTER 2 BACKGROUND

### 2.1 Distributed Cognition and Embodied Interaction

Distributed Cognition, developed by Edwin Hutchins, refers to a cognitive science theory that the people's knowledge not only lies within the individual but also in the individual's social and physical environment. The most apparent three kinds of distribution of cognitive process are *"...cognitive processes may be distributed across the members of a social group, cognitive processes may be distributed in the sense that the operation of the cognitive system involves coordination between internal and external (material or environmental) structure, and processes may be distributed through time in such a way that the products of earlier events can transform the nature of later events."*<sup>10</sup> Hutchins has described the essence of Distributed Cognition: *"As opposed to the world being in our mind, the mind is in the world."*<sup>11</sup> That is, the necessary knowledge and cognition to understanding the world does not existing solely within one's head, but distributed across objects, artifacts, and tools in the environment. Human conducts cognitive activities across distributed objects in the space in order to understand the environment and gather knowledge. The Distributed Cognition acts like a theoretical foundation for a large domain of knowledges, such as Tangible Interaction, Social Computing, and Embodied Interaction. In the Embodied Interaction model, people can not only interact with physical augmented items with computational functionality, which allows designers to create new sorts of metaphors, or take advantage of our physical skills, but also directly observe and respond to our physical activities in the world.

---

<sup>10</sup> Hutchins, E. "Cognition, Distributed." *International Encyclopedia of the Social & Behavioral Sciences*, 2001, 2068-072. doi:10.1016/b0-08-043076-7/01636-3.

<sup>11</sup> Hutchins, Edwin. *Cognition in the Wild*. MIT Press, 2006.

## 2.2 Internet of Things

As one of the practical domains which substantiated theoretical researches, the Internet of Things (IoT) is a concept which indicating a network of devices can connect, interact and exchange data. Human has the vision of machines connecting and communicating with each other since the early 1800s. As the first instantiation of the idea, Carl Friedrich Gauss and Wilhelm Weber invented their own communication protocol to communicate between electromagnetic telegraphs over a distance of 1200m in Gottingen, German.<sup>12</sup> With the development of the Internet, initialized as a part of the Defense Advanced Research Projects Agency in 1962, people were visioning to create universal connectivity upon every object. Karl Steinbach mentioned: *"In a few decades time, computers will be interwoven into almost every industrial product."*<sup>13</sup>

Late in the late 1970s, the concept of networking smart devices was firstly been discussed and attempted on a Coke vending machine (Figure 3) at Carnegie Mellon University<sup>14</sup>. As following "No real programmer can function without caffeine", the vending machine was very popular in the Computer Science Department. However, faculties and students wanted to know the inventory and coldness of the Coke so that they didn't need to travel a long way from the office to the machine but left with disappointment because of the Coke is out of stock or not cold enough. They set up a Finger server over the standard ARPANET protocols, installed micro-switches in the machine, and wrote a program to keep tabs on the Coke machine's state,

---

<sup>12</sup> "Gauss-Weber Telegraph – 1833." MagLab, [nationalmaglab.org/education/magnet-academy/history-of-electricity-magnetism/museum/gauss-weber-telegraph](http://nationalmaglab.org/education/magnet-academy/history-of-electricity-magnetism/museum/gauss-weber-telegraph).

<sup>13</sup> Mansell, Robin. *The Oxford Handbook of Information and Communication Technologies*. Oxford University Press, 2007.

<sup>14</sup> CMU SCS Coke Machine Home Page, [www.cs.cmu.edu/~coke/](http://www.cs.cmu.edu/~coke/).

including how long each bottle had been in the machine. After finishing all of these changes, people could check the machine from any CMU computer or from any machine anywhere on the Internet.



Figure 3: The CMU Coke Machine and its maintainers ,sources: Keynote presentation on the Internet of Things given by Paul Wilson, Director General at APNIC, at the inaugural Taiwan Internet Forum, held in Taipei, Taiwan from 8 December 2015

In 1998, the Internet Protocol Version 6 became a draft standard, which remarkably increased the Internet address space makes assigning an address to every object in the world becomes possible. The Internet of Things concept was officially named by Kevin Ashton in his talk in Procter & Gamble, in 1999. *“If we had computers that knew everything there was to know about things — using data they gathered without any help from us — we would be able to track and count everything, and greatly reduce waste, loss, and cost. We would know when things needed replacing, repairing, or recalling, and whether they were fresh or past their best. The Internet of Things has the potential to change the world, just as the Internet did. Maybe even more so.”*<sup>15</sup> At that time, He provided the idea of using the Radio-frequency Identification

---

<sup>15</sup> Journal, RFID. “That 'Internet of Things' Thing.” *That 'Internet of Things' Thing* - 2009-06-22 - Page 1 - RFID Journal, [www.rfidjournal.com/articles/view?4986](http://www.rfidjournal.com/articles/view?4986).

(RFID) in P&G's supply chain for tracking, which is considered to be one of the earliest practical IoT system implementations.

In 2002, Chana Schoenberger and Bruce Upbin published “The Internet of Things”, which introduces up-to-date deployments of IoT technologies from different companies. Chain retail club likes Sam’s Club attached electronic chip in the case of goods for automatic amount counting, replenishment detecting, and location tracking. As they stated in their article, “*Stores have eyes. Now they're getting brains. Soon tiny wireless chips stuck on shampoo bottles and jeans will track all that you wear and buy.*”<sup>16</sup> The development of the wireless chips provide people with endless imagination for creating a more ubiquitous environment, which, the ultimate goal, is to enable a more convenient environment.

When enterprises realized the potential profit that they can make by enabling the power of Internet of Things, designers and developers were also started to vision the Internet of Things techniques as a way to express their thinking and themselves. David Rose, an award-winning product designer, entrepreneur, and lecturer at the MIT Media Lab, co-founded Ambient Devices, which derived from the MIT Media Lab. He is also a pioneer in embedding Internet information in everyday objects. He mentioned in his book *Enchanted Objects: Design, Human Desire, and the Internet of Things*. “*I want the future of our relationship with digital technology to look less like the cold slab of glass of my nightmare and more like my grandfather’s basement workshop—chock-full of beloved tools and artifacts imbued with stories. I want the computer-human interface to be an empowering and positive experience—to minimize the interruption,*

---

<sup>16</sup> Schoenberger, Chana R. “The Internet of Things.” *Forbes*, Forbes Magazine, 19 June 2013, [www.forbes.com/global/2002/0318/092.html#35bc4ff73c3e](http://www.forbes.com/global/2002/0318/092.html#35bc4ff73c3e).

annoyance, and distraction of our so-called smartphones and glass-faced tablets.”<sup>17</sup> He envisioned the Internet of Things more like a way to express his considerations of how to transform the daily objects into the form he thinks them should be.



Figure 4: *The Enchanted Object* by David Rose, *Enchanted Objects* exhibit at the Boston Museum of Science, <https://enchantedobjects.com/ambient-devices-and-the-museum-of-science>

The Ambient Orb (Figure 4) was one of Rose's creations and a revolutionary ubiquitous computing device, providing real-time data to a user via a simple, color-changing interface. Ambient Umbrella (Figure 4) is another “enchanted object” which works by illuminating the handle of the umbrella if snow, thunderstorms, rain, or drizzle is in the forecast by embedded a wireless chip which connects to the forecast system’s API. Each weather has a different light

<sup>17</sup> Rose, David. *Enchanted Objects: Design, Human Desire, and the Internet of Things*. Scribner, 2014.

color so that users know what to expect. The activity of bringing the umbrella largely relies on the weather, but people still need to forwardly check the forecast before deciding to take the umbrella with them or not. David Rose is very sensitive to these kinds of object and activity relation and eagers to create solutions for bridging these relation gaps. Similar to the Ambient Umbrella and Ambient Orb, they also created other internet-connected objects including the Ambient Dashboard, 5-day Weather Forecaster, and the Energy Joule.

### **2.3 Interactive Architecture**

Derived from the idea of introducing Internet of Thing techniques to the architectural space, the Interactive Architecture refers to the interdisciplinary field which featuring the buildings with a series of sensors, processors and actuators, applied as the essential part of its functionality and characteristic. Interactive architecture encompasses building automation which as one of the practical targets for Internet of Things, but goes beyond it by including forms of interaction engagements and responses that may lay in pure communication purposes, synesthesia expression, as well as in the emotive and artistic field, thus related with the domain of interactive art. The scale of interactive architecture ranges from small interactive gadgets that deployed in the built space to large scale smart city and interactive landscape architecture, which makes it become feasible and applicable to many different design attempts. The design of interactive architecture is not like inventing new technology or extrapolating from it to fit an architectural purpose, but more like how to leverage all of these existing technologies and

represent the design thinking through these digital media and tangible interaction, which is more expressive and unambiguous compared to traditional 2D graphical representation.

LightSwarm (Figure 5), an interactive architecture project by Future Cities Lab in San Francisco, combined the lighting and acoustic factors and represents them to the occupants simultaneously. The existence of these factors helps the occupant to experience the space.

Michael Fox also has similar opinion that, *“We do not inhabit architectural space simply or shelter; we do so because we need the experience of space, which hinges on a great number of actors at once, including lighting, acoustics, materiality, and the other people in that space.”*<sup>18</sup>

The Future Cities Lab deployed acoustic sensors on the hallway and the facade of the building to capture the vibration from the pedestrian ‘s voice and steps and translated these data to visualize by a giant swam-like light installation composed by 450 individual LED modules. Sound transmitted the auditory pulse and magnified it into the representation of light, which translated the chaotic acoustic experience into visible and perceptible phenomenon. In this way, they enabled the occupants to “experience” the noise.

---

<sup>18</sup> Fox, Michael. *Interactive Architecture Adaptive World*. Princeton Architectural Press, 2016. p50



*Figure 5: LightSwarm, Jason Kelly Johnson, Nataly Gattegno, Ripon DeLeon Future Cities Lab, San Francisco, An sound-light effect interactive architecture project*

From another perspective, the interactive architecture is in a unique position that it repositions the role of designer, from aiming for creating a finished design to positioning the goal as catalyzing, which is probing how the design should evolve and designing the platform for the end-user to design. Within this context, the interactions with architectural spaces and objects become very important. Gestural language as one of the most intuitive means of control for enabling real physical interaction, has become the main consideration in many interactive architecture projects (Figure 6). Michael Fox and Allyn Polancic abandoned the direct sensing and response condition but was looking for more direct but natural control which is always being neglected by the limited sensing capabilities. They applied Microsoft Kinect sensor for tracking the gestures of the occupant (Figure 7). They envisioned the architecture should serve as an interface and allowed the user to directly interact with the environment via gestural language. The design of interactive architecture is no longer designing the final form but to create an initial state companying by constraints to enable the end-users to experience, explore and discover.

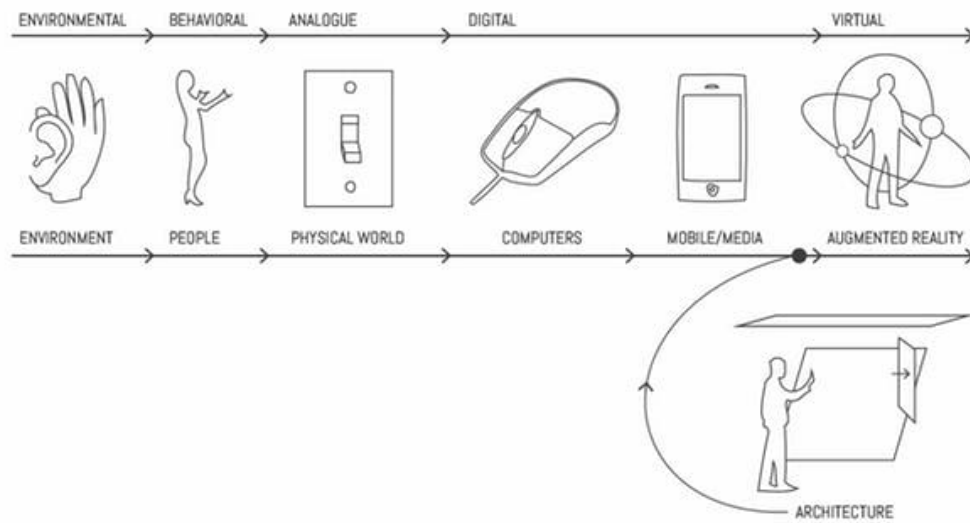


Figure 6: Evolution of Gesture Control, Michael Fox and Allyn Polancic, <https://archinect.com/allyn/project/conventions-of-control>



Figure 7: Testing environment setting, Michael Fox and Allyn Polancic, <https://archinect.com/allyn/project/conventions-of-control>

## 2.4 Multimodal Interactions

Although the gestural input has been explored in many interactive architecture projects, we are not living in a world only with gestures, which elicited the idea of introducing multiple modalities to the interactive architecture. Multimodal interaction exploits the synergic use of different modalities to optimize how users can accomplish interactive tasks. Sharon Oviatt mentioned in his paper *Ten Myths of Multimodal Interaction*, “Multimodal systems process combined natural input modes—such as speech, pen, touch, hand gestures, eye gaze, and head and body movements—in a coordinated manner with multimedia system output.”<sup>19</sup> Indeed, while the technology is rapidly developing, the way that human interacts with the machine also drawn a lot of people’s attention. The goal of presenting the interaction with multiple different modalities is for enabling a “more free and natural communication, interfacing users with automated systems in both input and output.”<sup>20</sup> Early systems such as ‘Put-that-there’ (Figure 8) by Richard Bolt from the Architecture Machine Group, MIT, allows a user simply point at a virtual object and literally tell the system to put that object somewhere else by pointing at a different location on a 2D projected display.

---

<sup>19</sup> Oviatt, Sharon. “Ten Myths of Multimodal Interaction.” *Communications of the ACM*, vol. 42, no. 11, 1999, pp. 74–81., doi:10.1145/319382.319398.

<sup>20</sup> Stivers, Tanya, and Jack Sidnell. “Introduction: Multimodal Interaction.” *Semiotica*, vol. 2005, no. 156, 2005, pp. 1–20., doi:10.1515/semi.2005.2005.156.1.



Figure 8: “Put-that-there”, Richard A. Bolt, Architecture Machine Group, MIT

“Quickset” (Figure 9), a research project from Oregon Graduate Institute of Science and Technology, demonstrated a military planning interface that allows a user to pen down at a location on a map and utter the name of a unit (e.g., “red T72 platoon”) to place at that location.

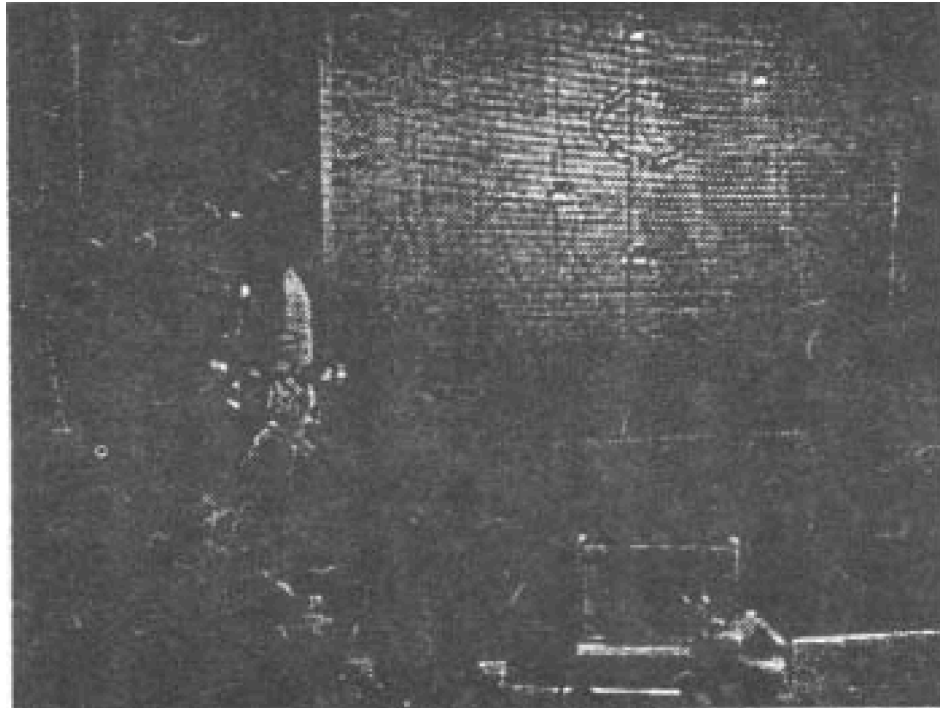


*Figure 9: QuickSet running on a wireless handheld PC.*

To optimally combine different modalities, it is important to understand their individual characteristics. In general, pointing and gestures were found to be more useful when specifying targets that are “perceptually accessible” to a user while speech is better at specifying “abstract or discrete actions.”<sup>21</sup> Oviatt et al. summarized that “basic subject, verb, and object constituents almost always are spoken, whereas those describing locative information invariably are written or gestured.”<sup>22</sup> Indeed, as Oviatt et al. pointed out, multimodal input “occurred most frequently

---

<sup>21</sup> Sun, Yong, et al. “A Novel Method for Multi-Sensory Data Fusion in Multimodal Human Computer Interaction.” Proceedings of the 20th Conference of the Computer-Human Interaction Special Interest Group (CHISIG) of Australia on Computer-Human Interaction: Design: Activities, Artefacts and Environments - OZCHI '06, 2006, doi:10.1145/1228175.1228257.

<sup>22</sup> Oviatt, Sharon, Phil Cohen, Lizhong Wu, Lisbeth Duncan, Bernhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson, and David Ferro. “Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions.” *Human-Computer Interaction* 15, no. 4 (12 2000): 263-322. doi:10.1207/s15327051hci1504\_1.

during spatial location commands”, which suggests the potential of multimodally specifying and interacting with interactive architecture objects situated across the physical space. To interpret multimodal input, it is important to understand the ‘interaction’ between modalities. Oviatt found that users’ natural language input is simplified during multimodal interaction, as the complexity is offloaded to more expressive modalities. Enabling these modalities in the interactive architecture context provides real-time characteristic for designers to build up the interactive platform for catalyzing the creativities. As Gordon Pask states in his foreword to *An Evolutionary Architecture*: “*The role of the architect here, I think, is not so much to design a building or city as to catalyze them: to act that they may evolve.*”<sup>23</sup>

## 2.5 End-user Development

Connected objects, serve as the appliances in interactive architecture, are very crucial to the environment. Empowering the communication and calculation capabilities, the tiny computer plays as the brain for the object and talk to each other via the Internet. Single-board microcontrollers are the hardware level infrastructure for the devices. They are integrated with the microprocessor, I/O circuits, clock generator, RAM and other ICs to provide all circuitry necessities for control tasks. However, programming on single-board microcontroller requires great domain knowledges, developers have to conduct memory flashing or hex keypad monitoring, none of which is user-friendly for end-users, beginners, or the designer who interests in creating reactive object for the interactive architecture project.

---

<sup>23</sup> Frazer, John. *An Evolutionary Architecture*. Architectural Association, 1995.

Later on, researchers were trying to add one more layer of abstraction on top of embedded system programming. Programming interface such as Arduino Hardware (Figure 10) and Arduino IDE (Figure 11) was designed for providing clean and straightforward coding interface for users, and the interface will handle all of the complex data transaction protocol and program configuration processes in the background. It only requires the user has basic I/O and control flow knowledge to define the behavior of the hardware. However, it is still assuming the user has programming skill and also requiring a connection between the hardware to host devices (PCs).

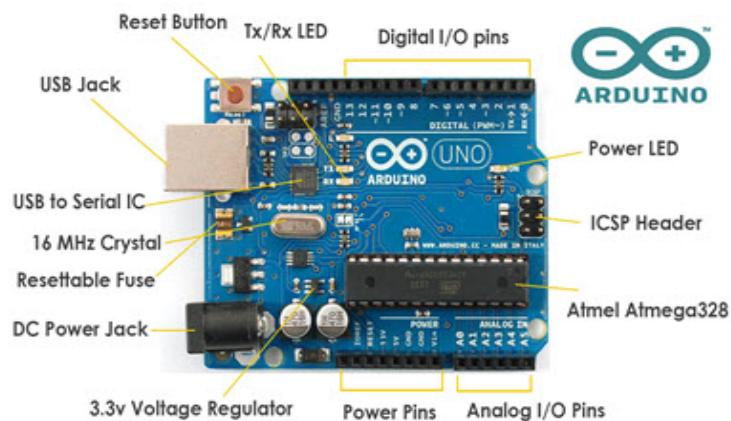


Figure 10: Arduino UNO layout, <https://www.arduino.cc/>



Figure 11: Arduino IDE interface

To solve these drawbacks, on one hand, researchers and developers were trying to build up another level of programming abstraction, Mazzei Et al. developed a web-based IoT device design platform which includes abstracted functional toolboxes which user can drag and drop to define the device's behavior without typing a single line of code.<sup>24</sup> The NodeRed (Figure 12) is a flow-based visual programming tool for connecting hardware, online services, and APIs which also reduce the degree of dependency of professional domain knowledges presented by IBM Emerging Technology group. *"It is a model that lends itself very well to visual representation and makes it more accessible to a wider range of users. If someone can break down a problem into discrete steps they can look at a flow and get a sense of what it is doing; without having to understand the individual lines of code within each node."*<sup>25</sup> The extensibility and efficiency of NodeRed allow the user to focus on designing the logical flow and leave the dirty works to the platform.

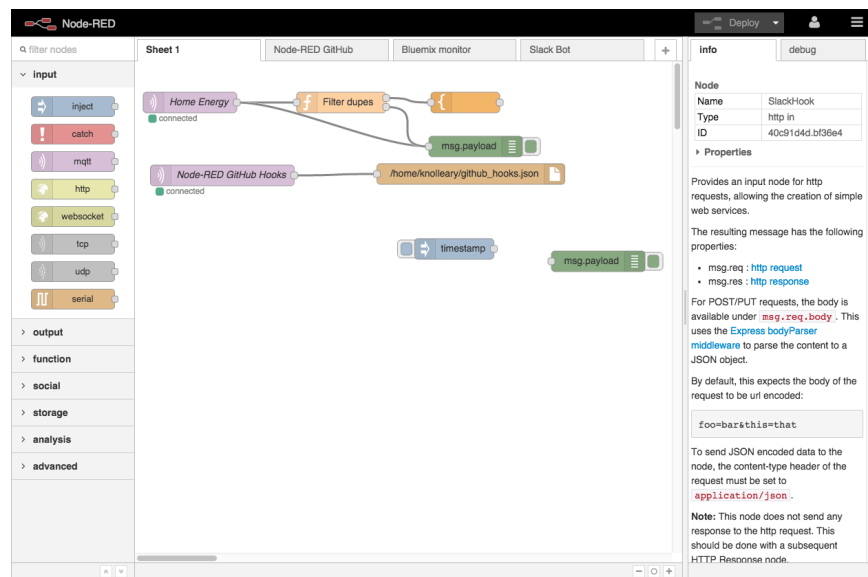


Figure 12: The NodeRed interface

<sup>24</sup> Mazzei, Daniele, Gualtiero Fantoni, Gabriele Montelisciani, and Giacomo Baldi. "Internet of Things for Designing Smart Objects." *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 03 2014. doi:10.1109/wf-iot.2014.6803175

<sup>25</sup> NodeRed. <https://nodered.org/>.

The NodeRed system is useful in sense of easing the configuration efforts and showing the hierarchy of the program while also combining the hardware, online services, and APIs. However, because of the complexity of the system, it still requires the end-user to have deep knowledge of the logical flow and understand the meaning of each component. As Cvijikj Et al mentioned, “*The development of tools and techniques is a key focus for the concept of participatory design (PD). End-user tools and techniques should promote a practice where researchers and design professionals will be able to learn about users’ work, and where users will be able to take an active part in technology design.*”<sup>26</sup> They proposed that the tools should avoid traditional abstraction of design techniques but allow users easily experiment with a variety of design possibilities but still in an efficient way.

These relative domains covered the background of the research in order to provide insights to elicit the design and the implementation detail of the research.

---

<sup>26</sup> Cvijikj, Irena Pletikosa, and Florian Michahelles. "The Toolkit Approach for End-user Participation in the Internet of Things." *Architecting the Internet of Things*, 2011, 65-96. doi:10.1007/978-3-642-19157-2\_4.

## CHAPTER 3 RESEARCH DESIGN AND METHODOLOGY

In this chapter, the research and the implementation were guided by the three sub-questions (see Chapter 1) that arose from the main research question. A formative study was conducted to understand the key capabilities that the interactive system should have; then a backend system and a hardware sensing platform was developed to support both creative and practical activities; Finally, a file-based configuration format and lightweight schema for rule-based interactions were prepared to ease the difficulty of designing reactive appliances and provide increased convenience for end-user development and rapid prototyping.

### 3.1 Preliminary Formative Study

To inform the creation of design space and system implementation, also for understanding what the intuitive and suitable way for the user to interact with interactive architecture space, I conducted a formative study to investigate the sensing capabilities and user-initiated interaction interface for the interactive architectural system. The study was conducted in August 2018 in CodeLab, Carnegie Mellon University.

I recruited 10 participants (aged 18-28, 6 males, 4 females). Five were native English speakers; eight had experience interacting with intelligent voice assistants like Google Home or Amazon Alexa. Figure 13 shows an arrangement of eight typical appliances around a user in a  $6\text{m} \times 10\text{m}$  space. I chose these appliances so that there were both single (e.g., one Roomba, one projector, one speaker) and multiple (e.g., four lights) devices with various tasks described below for covering the differences in the amount of the appliance which would possibly influence the way of interaction. Each participant was not required to wear any device.

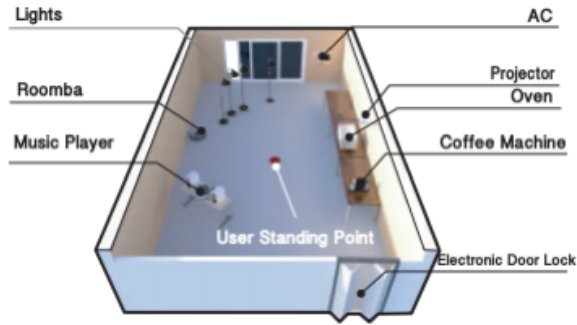


Figure 13: The layout of the study environment

Participants were asked to remotely provide indication for interact with the appliances while standing at the center of the public space of CodeLab. For task design, I considered the distinction between commands (e.g., turn on/off, change volume, clean an area) and information (e.g., when will the coffee be ready, what is the AC’s set temperature). To avoid priming the participants, I described each command-task by showing the participants a pre-recorded video of an appliance state change (e.g., a lamp going from off to on). Participants were asked to come up with their own preferred input by the question: “Assuming all appliances are listening to you, and there is a system that can understand whatever you do, how would you remotely cause the appliance(s) into the effect shown in the video?” The studies were video recorded, and participants’ comments and responses during the tasks were also gathered.

From the study, I found that the way participants expressed their intent of interacting with devices almost always consisted of two components: Selection—specifying a target device, and Interaction—specifying the command or requesting information. However, participants’ preferences varied in using gestures and/or voice as their desired interaction modality. For selection, eight of the ten participants moved their arm towards the target appliance, although their specific hand gestures differed (pointing, circling, tapping, waving). For interaction (specifying commands or requesting information), six participants interacted with appliances

multimodally (i.e., using both gestural and verbal commands) while one participant chose to use gestures only and three others used voice only.

The finding of the study also aligns with the insights of Paul Dourish. As he mentioned in his book *Where the Action is, The foundations of embodied interactions*.<sup>27</sup> “... *Similarly, when speaking in a normal, everyday way, one doesn’t have to rationally plan and control each word while saying it. The same is true for the other utterances, gestures and symbols–in–use that collectively constitute everyday activity.*” He implicated that the intuition of interaction comes from the activities that people conduct every day. He called these familiarities as “*Present-at-hand*” which takes shape from the previous experiences, “*For example, creativity can be considered as the variation of an individual’s subjective understanding from his or her prior understanding and from others’.* *The individual may then be very conscious of his or her own activity, rationalizing it and very aware of it, ... With experience of its use, however, it may become understood and familiar.*”<sup>28</sup>

From the study and the insights of expert, the findings implicated that, first of all, the interactive architecture sensing system should be able to tackle the two aspects of the user-initiated interaction - selection and interaction. Secondly, the spatial awareness is the most essential capability for the interactive architecture project. It provides the context for the user to experience the space, and allows the system to aware of the movement, activities, and status of the user, from which the information could be exploited as the resource for the reaction of the interactive appliances that controlled by the interactive system. Last but not the least, the verbal and gestural commands should be considered as interchangeably and collaboratively. Due to the

---

<sup>27</sup> Dourish, Paul. *Where the Action Is the Foundations of Embodied Interaction*. Bradford Books, 2004.

<sup>28</sup> Dourish, Paul. *Where the Action Is the Foundations of Embodied Interaction*. Bradford Books, 2004.

fact that people communicate the each other every day, the voice and gestures work as the main mediums for human to human interaction and has already been familiarized from everyday activities. Enabling the similar interaction capabilities for the interactive space would help the user for easier and better understanding and interactions.

### 3.2 System Design and Implementation

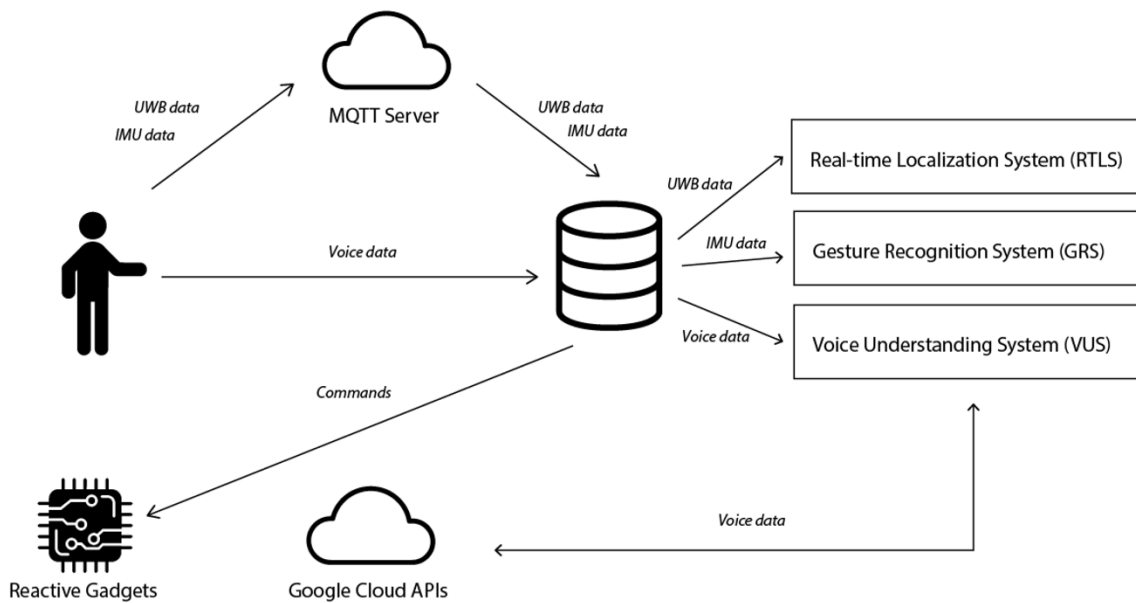


Figure 14: System Structure

Following the preliminary formative study, the interactive sensing system should have the capabilities to understand the spatial context and the real-time user information, and also be able to response to both gestural and verbal commands. In another words, the three main functional components of the system are the real-time localization system, gesture recognition system and the verbal command processing system. To enable more direct and timely data collection, instead of implementing a computer vision approach, the interactive system takes an on-body sensor approach which requires the end-user to wear a hardware sensing wristband, that

integrated the sensors and a microcontroller, the structure of the whole system is shown in Figure 14.

### **3.2.1 Centric Backend System**

#### ***3.2.1.1 Real-Time Localization System (RTLS)***

For allowing the system to have spatial awareness, I implemented an Ultra-wideband (UWB) based Real-Time Localization tracking System (RTLS). Ultra-wideband is a radio technology that communicates with high-bandwidth low energy level of radio spectrum in order to avoid the inference to maintain high accuracy. Before selecting UWB as the primary localization engine, I horizontally compared the performance and fitness of different RTLS techniques. For indoor navigation purpose, localization system can be spread into video-based (camera and computer vision) and trilateration-based. The video-based localization engine detects the feature points and the descriptors to translate the real-world coordinate to the camera coordinate by using homogeneous transforms. However, due to the amount of data existing in each image frame of the video, video-based localization technique requires high computational

capability. Moreover, it also draws privacy issues because of the lively processing of supervisory camera videos.

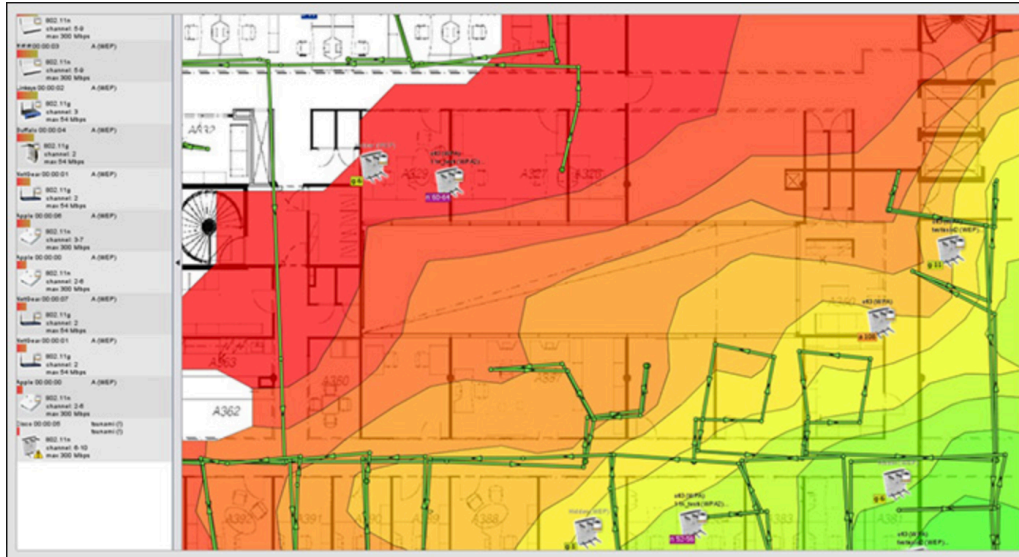


Figure 15: A signal strength heat map for RSSI based localization approach, source: Ekahau HeatMapper, <https://www.ekahau.com/products/heatmapper/overview/>

On the other hand, trilateration-based localization technique is more lightweight and timelier for figuring out the location of the user within the space because it calculates the location through indirect distant data. The existing technical stack of implementation of trilateration-based localization system includes BLE, Wi-Fi, Ultrasound, and Ultra-wideband. BLE and Wi-Fi are based on indirect hints for localization, which can also be referred to as the Received Signal Strength Indicator (RSSI). The system will construct a signal strength bump map and speculate the possible location of the target (Figure 15). Although they are easy to measure, they suffer from the fact that they need to heavily rely on the environment. Because of this environmental dependency, the results are hard to be very accurate and consistent. For the Ultra-sound and Ultra-wideband techniques, the triangulation for localization is calculated from the distances of the anchors to moving tags, which can also be inferred as Time of Flight (ToF) solution. (Figure 16) As long as the system have at least three distances to the location-known

anchors, it can calculate the location of the tag mathematically. Compare to the RSSI attempts, although the ToF solution requires more hardware settings, it has far better accuracy and reliability, which are very crucial for my system.

To increase the accuracy of the RTLS, I also applied a Kalman Filter<sup>29</sup> on the real-time localization data. The Kalman Filter has implemented as the optimal solution for many tracking and data prediction tasks. The Kalman Filter (eq. 3.1) requires a Kalman Gain  $K_k$  to manipulate the confidence on the current measured value  $Z_k$  and to take the previous estimation state  $\hat{X}_{k-1}$  into account for calculating the current estimation value  $\hat{X}_k$ . The UWB sensor localization engine provide an accuracy factor which is calculated by its maximum likelihood estimation and I used it as the Kalman Gain. The system takes previous localization data as the estimation and uses the current measured value with the Kalman Gain to calculate the current estimation.

$$(eq. 3.1). \quad \hat{X}_k = K_k \cdot Z_k + (1 - K_k) \cdot \hat{X}_{k-1}$$

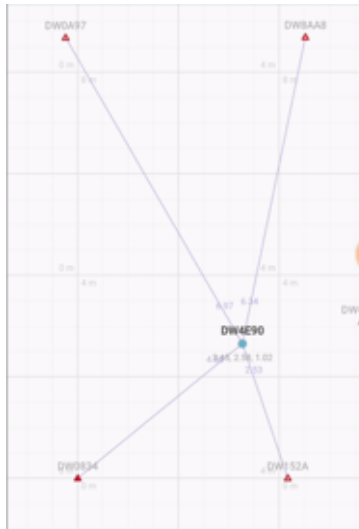


Figure 16: UWB ToF based localization, visualization from DecaWave UWB Software

<sup>29</sup> Kalman, R. E. "A New Approach to Linear Filtering and Prediction Problems." *Journal of Basic Engineering* 82, no. 1 (1960): 35. doi:10.1115/1.3662552.

For enabling seamless data transaction and communication, I run an MQTT<sup>30</sup> broker on a local Raspberry Pi (Figure 17). MQTT is a popular lightweight protocol for IoT data communication which implements the publish/subscribe model. All of the data, commands, and messages will be published to the corresponding MQTT topic, the MQTT has message queue for handling the requests and response for producers and consumers.

I selected to use the DecaWave DWM1001 Ultra-wideband sensors as the ToF trilateration-based real-time localization system. DecaWave provided out-of-shelf localization engine and expose different APIs for data retrieving. However, they only provided the Raspberry Pi API and Linux based C programming API in their current version, it is not compatible with the Arduino-based ARM M3 microcontroller implementation. I wrote my own version of API by translating the bytes sending sequence by Arduino based library.

All of the end-user localization data will be sent to the “*data/userLoc*” topic under the MQTT broker, the backend system subscribe the topic and can be notified whenever the data is arrived. The protocol of the localization data is as following:

*“Xcoordinate\_Ycoordinate\_Zcoordinate\_userIdentifier”*

When the data arrived, the backend system uses the identifier for figuring out the location data belonging, and then parses and applies the 3-axis coordinate to the corresponding user model.

---

<sup>30</sup> "MQTT." MQTT RSS. <http://mqtt.org/>.

The image shows a Raspberry Pi terminal window titled 'Terminal 1: mosquitto (121x29)' displaying a log of MQTT broker activity. The log includes messages such as 'Received PINGREQ from ESP8266 Client', 'Sending PINGRESP to ESP8266 Client', 'New connection from 192.168.1.246 on port', 'New client connected from 192.168.1.246 as', 'Sending CONNACK to paho/8F54895D50DC295814', 'Received PUBLISH from paho/8F54895D50DC295814', 'Sending PUBLISH to ESP8266 Client (d0, q0, 1)', 'Received DISCONNECT from paho/8F54895D50DC295814', 'Client paho/8F54895D50DC295814 disconnected', 'Received PINGREQ from ESP8266 Client', 'Sending PINGRESP to ESP8266 Client', 'Received PINGREQ from ESP8266 Client', 'Sending PINGRESP to ESP8266 Client', 'Received PINGREQ from ESP8266 Client', 'Sending PINGRESP to ESP8266 Client', 'Received PINGREQ from ESP8266 Client', 'Sending PINGRESP to ESP8266 Client', 'New connection from 192.168.1.246 on port', 'New client connected from 192.168.1.246 as', 'Sending CONNACK to paho/F0C68457D925231A90', 'Received PUBLISH from paho/F0C68457D925231A90', 'Sending PUBLISH to ESP8266 Client (d0, q0, 1)', 'Received DISCONNECT from paho/F0C68457D925231A90', 'Client paho/F0C68457D925231A90 disconnected', 'Received PINGREQ from ESP8266 Client', and 'Sending PINGRESP to ESP8266 Client'.

Overlaid on the terminal is a Python script window titled 'mqtt\_example.py' with the following code:

```
1 import paho.mqtt.publish as publish
2
3 print("Sending 1...")
4 publish.single("ledStatus", "1", hostname="macman")
```

Below the script is a serial terminal window titled '/dev/cu.SLAB\_USBtoUART' showing the output of the script:

```
Message arrived [ledStatus] 1
Message arrived [ledStatus] 1
```

The serial terminal window also shows a 'Send' button and a status bar indicating 'Line 3, Column 22', 'Tab Size: 4', and 'Python'.

Figure 17: Mosquitto MQTT broker Raspberry Pi terminal interface

### 3.2.1.2 Gesture Recognition System (GRS)

Using the accelerometer and gyroscope data for detecting movement and gestures have been explored by many researchers. Susi et al.<sup>31</sup> implemented an adaptive threshold-based motion mode recognition and step detection system. However, although the prior work has explored thresholding the magnitudes of accelerometer and gyroscope to detect the onset of pointing and other gestures, the problem is that the high threshold value requires a user to extend their arm very fast in order for the pointing to be recognized.

<sup>31</sup> Susi, Melania, Valérie Renaudin, and Gérard Lachapelle. "Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users." *Sensors* 13, no. 2 (01, 2013): 1539-562. doi:10.3390/s130201539.

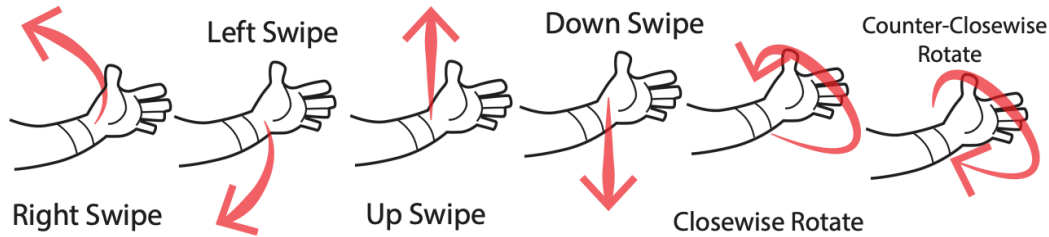


Figure 18: The default supported gestures

To overcome this problem, I take a data-driven approach: as a pointing gesture is performed, I collect raw data from the accelerometer and gyroscope over an empirically defined 1.5-second window. An application was writing for providing the end-user with a graphical training interface (Figure 21). By specifying what gesture, you are presenting and press the start button and then perform the gesture, the application will automatically record the data of the accelerometer and gyroscope data and label it corresponding to what you indicated before. I trained a Random Forest classifier by the Weka (Figure 20) software which provided by University of Waikato, New Zealand, over 160 examples of pointing gestures (varied in directions and speeds), 150 samples of default control gestures (up-swipe, down-swipe, left-swipe, right-swipe, clockwise cycle, counter-clockwise cycle) (Figure 18) and 900 samples of inactivity (e.g., walking around, hand waving, typing on a computer, reading books). The Random Forest classifier achieved 98.59% in 10-fold cross-validation. (Figure 19) My approach runs a sliding time window to ‘pick up’ a user’s gesture, producing only one false positive over a collective 45 minutes of inactivity. The supported gestures aggregative elicited from participants’ input in the formative study.

==== Confusion Matrix ====

|    | a   | b  | c   | d  | e  | f    | g |                   |
|----|-----|----|-----|----|----|------|---|-------------------|
| 70 | 0   | 1  | 3   | 0  | 0  | 0    | 9 | a = leftSwap      |
| 0  | 114 | 0  | 0   | 0  | 0  | 0    | 6 | b = rightSwap     |
| 2  | 0   | 99 | 0   | 0  | 0  | 0    | 6 | c = upSwap        |
| 0  | 0   | 0  | 148 | 0  | 0  | 0    | 4 | d = downSwap      |
| 0  | 0   | 0  | 0   | 59 | 0  | 1    |   | e = circleCW      |
| 0  | 0   | 0  | 1   | 0  | 58 | 0    |   | f = circleCCW     |
| 0  | 0   | 0  | 0   | 0  | 0  | 1754 |   | g = noInteraction |

Figure 19: The confusion matrix of the gesture training

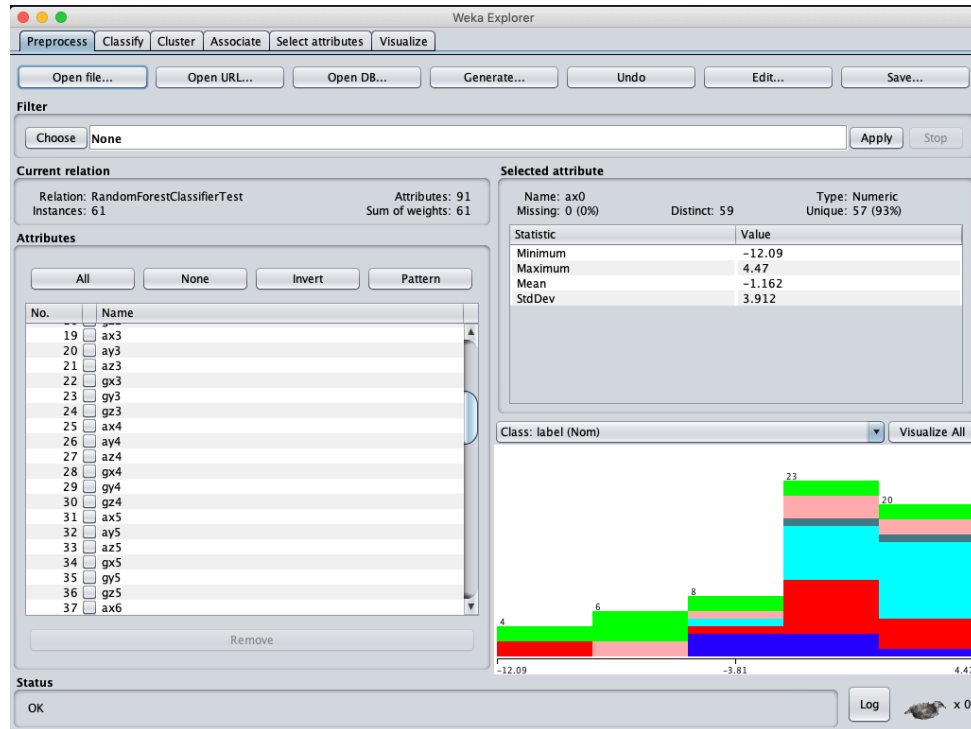


Figure 20: The machine learning interface of Weka

The recognition module is running independently in the backend system. The end-user accelerometer and gyroscope will stream the data to the “data/raw\_acc\_gyro” topic and also subscribed by the backend system. The protocol of the data is as following:

“accX\_accY\_accZ\_gyroX\_gyroY\_gyroZ\_userIdentifier”

The accelerometer and gyroscope data will be passed to a data parser to rearrange into the format that can be recognized by the model. If the system recognizes a gesture from the data, it

will send a message to the “data/gesture” in “gesture\_userIdentifier” format. The recognition is running in every 0.1 second for tracking the gesture in real time.

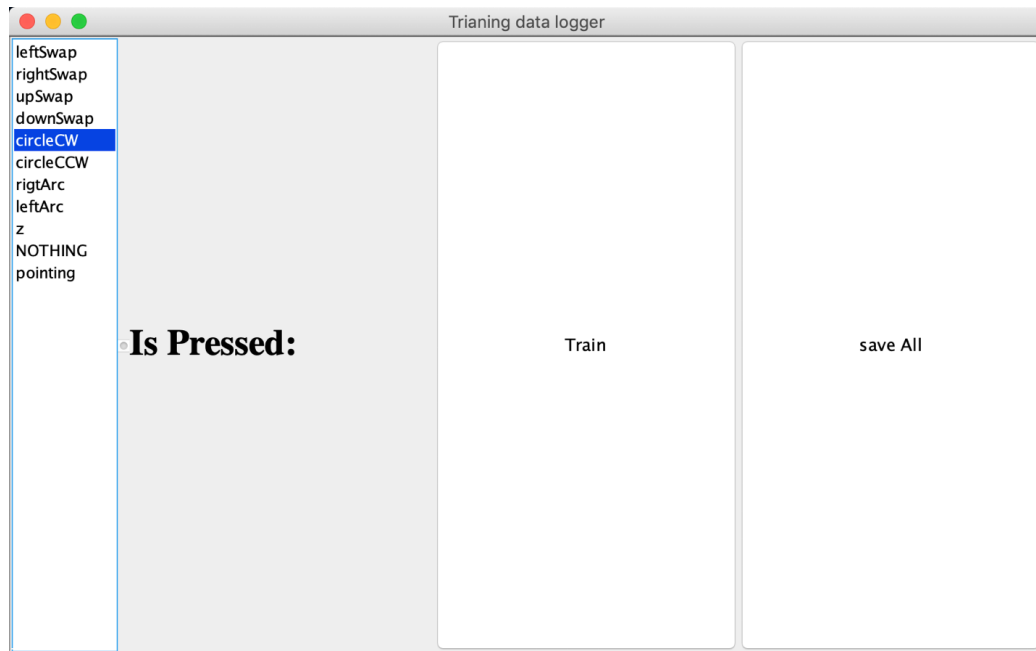


Figure 21: The training interface of my system

### 3.2.1.3 Voice Understanding System (VUS)

The voice understanding system is in a pipeline sequence of Speech-to-text and Natural language processing. Instead of adding one more speech taking component on the end-user device, which will increase the load and complexity of the system, I decided to direct the user audio input to the cloud API. All of the user audio data is collected by a BLE wireless mic and streams to the Google Cloud Platform Speech-to-Text API<sup>32</sup> and Natural Language API<sup>33</sup>. I

<sup>32</sup> "Cloud Speech-to-Text - Speech Recognition | Cloud Speech-to-Text API | Google Cloud." Google. <https://cloud.google.com/speech-to-text/>.

<sup>33</sup> "Cloud Natural Language | Cloud Natural Language API | Google Cloud." Google. <https://cloud.google.com/natural-language/>.

implemented my own data structure for representing the hierarchy of the verbal result. When a voice command is received, the system will construct the linguistic dependency graph and compare the parsed dependency graph with the selected device's repertoire of executable verbal commands' dependency graphs, which was constructed from the corpus collected in the formative study.

For example, the comparison will match “*follow the user*” to “*follow user*” by the DOBJ term “*user*” while considering the synonymous ROOT phrases ‘*follow*’. The comparison will also match “*increase the volume*” to “*raise the volume*” by the DOBJ ‘*volume*’ while considering ROOT phrases ‘*increase*’ and ‘*raise*.’ (Figure 22) If any of the verbal command is recognized, the backend system will inspect if the command being executable by the target appliance, if it is not, the executable frame will remain alive for 8 seconds until being destroyed, which is a typical approach used by commercial voice products.<sup>34</sup>

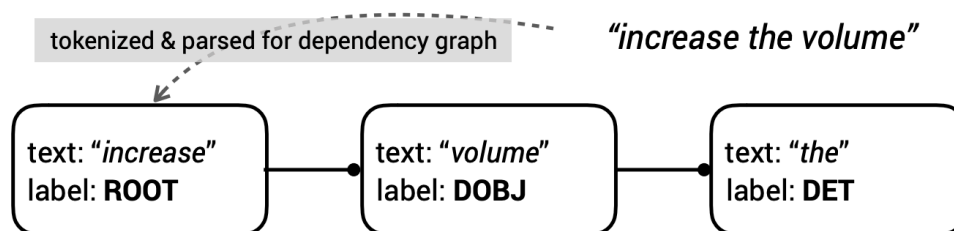


Figure 22: An example of the dependency parsed tree

<sup>34</sup> "Amazon.com Help: Echo Dot Help." Amazon.  
[https://www.amazon.com/gp/help/customer/display.html/ref=hp\\_200127470\\_echo\\_dot\\_img?nodeId=202010090](https://www.amazon.com/gp/help/customer/display.html/ref=hp_200127470_echo_dot_img?nodeId=202010090).

### 3.2.2 End-user Hardware Wristband and Appliances

#### 3.2.2.1 End-user Hardware Wristband

To set up the UWB real-time localization system, four UWB anchors should be attached on the walls around the room in different heights. Before integrating the UWB tracking tag into the system, I perform a technical evaluation for testing the localization performance of the UWB localization. By labeling a test grid on the floor with each intersection being measured the ground truth coordinates by a laser measurer, I stood on those intersections and compared the reading from the UWB localization system with the ground truth measurement for 600 data points. The overall mean localization error across all test points was 0.330m (SD = 0.169m).

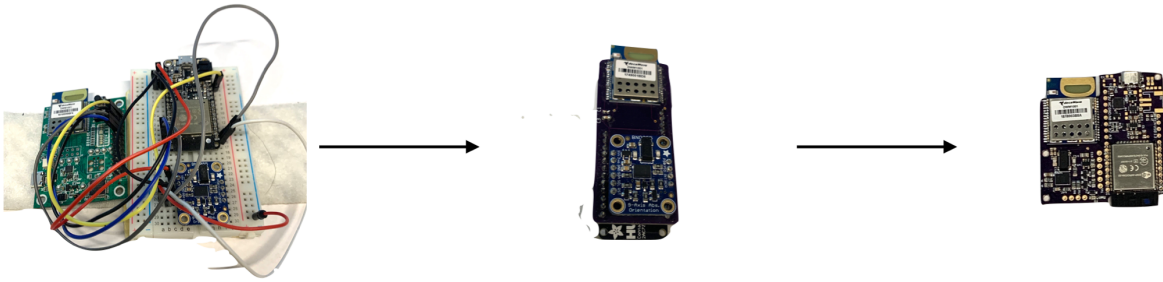


*Figure 23: The iteration of the hardware prototypes*

The structure of the end-user wristband (Figure 24 B) is comprised of an Espressif ESP32 Wireless MCU, a DecaWave DWM-1001 ultrawide-band unit and a Bosch BNO055 Inertial Measurement Units. The UWB unit is communicating via UART to the ESP32 control board which is Wi-Fi and BLE enabled. The update rate of the UWB localization is 10Hz, which is enough and able to support multiple users simultaneously. Before selected the ESP32 as the main calculation unit, I tested the other MCU like Arduino Yun and ESP8266, but they all turned out to be unsuitable for this situation because of lacking extra hardware serial ports. Due to the high

baud rate requirement (115200 bit/s) for communication with DWM1001 UWB unit, it is impossible to use the GPIO pins with software serial library to stably communicate in such a high speed. The Bosch BNO055 IMU units communicate with the MCU via SPI communication in 50 Hz frequency.

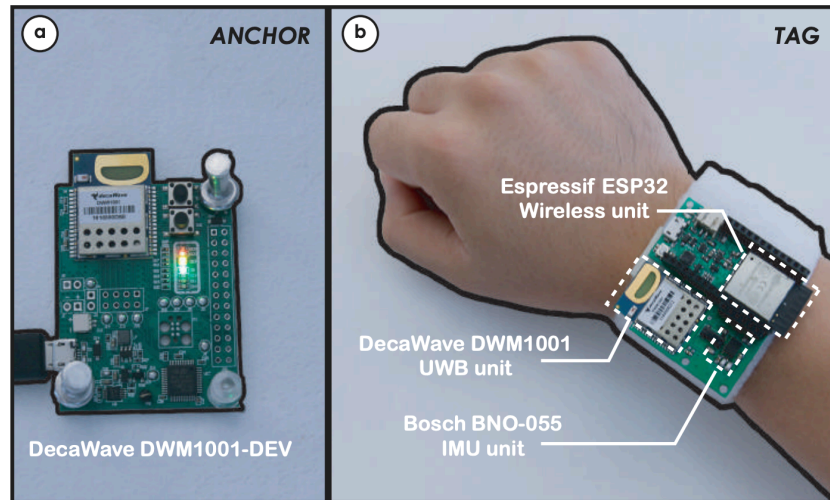


Figure 24: The anchors and end-user wristband

The logic of the MCU is continuously sending the accelerometer and gyroscope data from the IMU to the MQTT server. If a pointing gesture is recognized by the backend system, the MCU will be notified also via MQTT. Then the MCU will construct a localization data, orientation data and its identification to the corresponding MQTT topic in

*“Xcoordinate\_Ycoordinate\_Zcoordinate\_Pitch\_Yaw\_Roll\_userIdentifier”*

format. This implementation not only decouples the backend and the user-end hardware, but also migrates most of the calculation to the backend system which is running on powerful modern server to mitigate the computation load of the user-end hardware. Moreover, due to the fact that the system uses unique identifier to handle the affiliation of data, it makes the system to have the ability to server multiple end-user devices simultaneously, which solved the scalability problem that struggled many existing interactive architecture project systems.

For providing a visual feedback to the user when they triggered connection to the target appliance, I exploited the onboard LED of the ESP32 module, which is controlled by the GPIO pin #1. When a connection is established, the backend system will send a flag to the “*msg/triggered*” MQTT topic, which is subscribed by the ESP32 module LED. In this way, the end user will be able to distinguish if he/she successfully connected to the appliance.

After testing the individual sensor performance and the integrated performance on the breadboard, I designed and printed a PCB board to get rid of wirings for stability purpose. (Figure 23)

### ***3.2.2.2 Commercial Smart Appliances Reconfiguration***

The system has the capability to allow the end-user to register and configure any Internet enabled device into the system. However, the only prerequisite is the device should be MQTT enabled as well. Most of the existing commercial smart products is enabled with Wi-Fi but there are still a small number of products don't support MQTT. The commercial products that I've tested are the Sonoff B1<sup>35</sup> bulb and Sonoff R2<sup>36</sup> BASIC switch (Figure 25). Sonoff provides their own version of firmware which allows the end-user to download app to control the device, which is not what we are looking for here.

---

<sup>35</sup> Sonoff B1, <https://www.itread.cc/sonoff-b1.html>

<sup>36</sup> Sonoff R2, <https://www.itread.cc/sonoff-pow-r2.html>



*Figure 25: The tested commercial products*

Theo Arends from Netherlands posted a customized firmware named Tasmota<sup>37</sup> which is a MQTT firmware for the ESP8266 based devices. Since the Sonoff product are all manufactured upon the ESP8266 MCU, which means that I can flash the Tasmota firmware to enable its MQTT connection. To flash the firmware on the Sonoff products, I used an UART-TTL serial cable and soldered male pins to the GND, VCC, RX, TX ports of the Sonoff main board (Figure 26). When the firmware is successfully flashed, a graphical interface is provided for configuration of the device, which means that you can configure corresponding MQTT topic to control the function of the device. For other commercial device, if the device enabled MQTT control by default, the end-user only needs to send the correct commands to the corresponding topic; if it doesn't, using the Tasmota firmware will also be a feasible substitution for solving the problem.

---

<sup>37</sup> Arendst. "Arendst/Sonoff-Tasmota." *GitHub*, 25 Apr. 2019, [github.com/arendst/Sonoff-Tasmota](https://github.com/arendst/Sonoff-Tasmota).

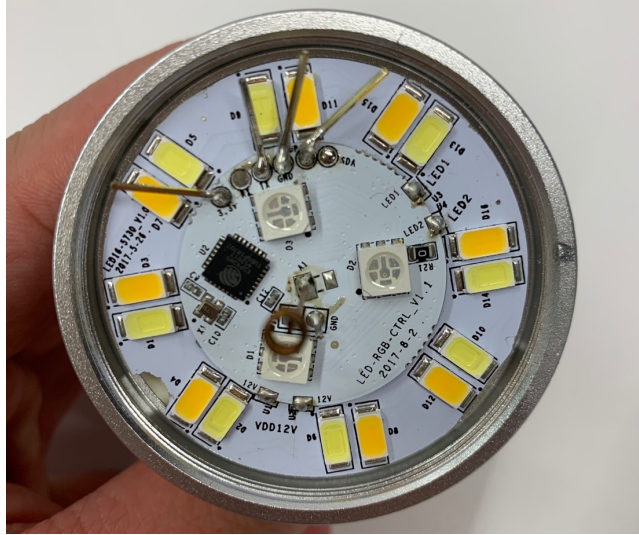


Figure 26: The layout of the Sonoff B1

### 3.2.2.3 Customized Smart Gadget Reconfiguration

Arduino provides a user-friendly interface and hardware kits that allows the user to DIY without having to much domain knowledge of hardware, which makes it the first choice for designers to create reactive gadgets and small devices in their interactive architecture projects. Considering this circumstance, my system also allows the users to register customized Arduino based gadget into the system (Figure 27). MQTT is the only dependency for the implementation, which the end-user can download the PubSubClient.h library<sup>38</sup> serves as the MQTT client library for C language-based Arduino boards by Nick O'Leary. Following the PubSubClient library example instruction, the only thing that the designer needs to configure is specifying what topic the board is listening to and what topic the board is going to publish on. The main logical implementation still remains the same as the normal interactive gadget.

---

<sup>38</sup> "Arduino Client for MQTT." Arduino Client for MQTT, [pubsubclient.knolleary.net/](https://pubsubclient.knolleary.net/).

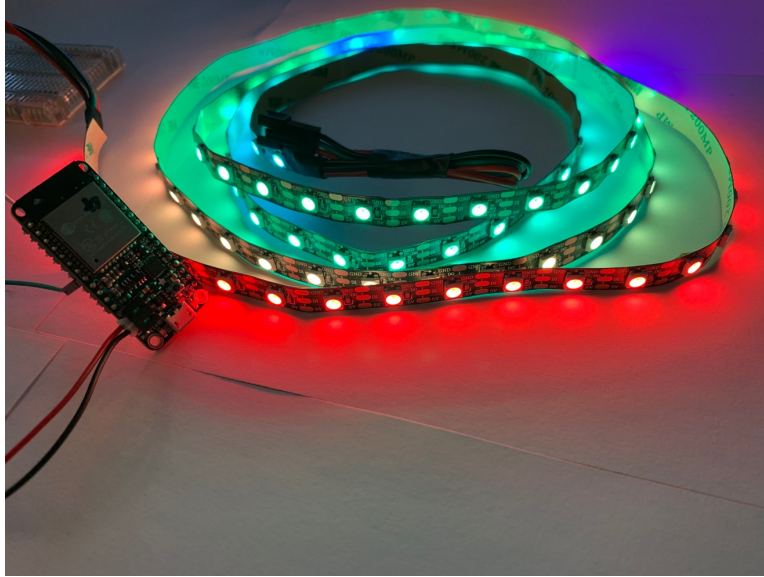


*Figure 27: A customized LED strip which controlled by the ESP32 board with PubSub library*

### **3.2.3 End-user Template and Configuration File**

After successfully implementing the system, for bridging the gap between the appliances configuration to the system. I created YAML format-based configurators which presents as configuration files. The end-user only need to type down the information of the appliances that they want to enable with the system, and the Rules that configure the extra behavior of the appliance in the corresponding configuration files, the system will automatically generate the virtual appliance objects into the system and apply rules to the appliance.

#### ***3.2.3.1 End-user Appliance Template and Configuration File***

The appliance is presented as Java Objects in the model of the system. So besides of registering the appliance into the network, it is also necessary to create the correct Java object in the backend system. For easing the process of generation and configuration, the backend system provides a parser which can generate object according to what the user indicated in the Appliance configuration YAML file. The user needs to provide the name, appliance type, the

MQTT topic, and spatial location of the object in the “appliance.yml” file. (Figure 28) The backend system provided a handful of default appliance classes that are considered as the Type configuration for the appliance. If the user wants to create their own Type for their customized gadget, the backend system exposes Appliance.java as the abstract template (Figure 29) which can be extended by the user customized class. The abstract class requires the user to override two functions, one is the commands that the appliance will send to the MQTT topic when an executable verbal command is received, the other is for the reaction when executable gestural command is received. The supported gestural commands and verbal commands also need to be specified in the constructor of the class.

```

1  appliance:
2    # an example of commercial product
3    - name: light1
4      type: Light
5      topic: cmdnd/sonoff1/POWER
6      locationX: 5800
7      locationY: 0
8      locationZ: 2390
9
10   - name: soundModule
11     type: SoundModule
12     topic: music
13     locationX: 5000
14     locationY: 5000
15     locationZ: 1000
16
17   - name: lightSoundStrip
18     type: LightSoundStrip
19     topic: SoundLight
20     locationX: 5700
21     locationY: 1000
22     locationZ: 1000
23
24
25   # an example of customized appliance
26   - name: lightStrip
27     type: LightStrip
28     topic: lightStrip
29     locationX: -1000
30     locationY: 1000
31     locationZ: 1000
32
33

```

APPLIANCE NAME  
APPLIANCE TYPE  
MQTT TOPIC  
X COORDINATE  
Y COORDINATE  
Z COORDINATE

Figure 28: The appliance configuration file

```

1  package model;
2
3  import java.util.*;
4
5
6
7
8
9
10 public abstract class Appliance {
11     private String name;
12     private Vector3D location;
13     private String topic;
14     protected Set<TokenNode> executableWords = new HashSet<>();
15     protected Set<String> supportedGestures = new HashSet<>();
16     protected Set<String> rootSet = new HashSet<>();
17
18     public Appliance(String name, Vector3D location, String topic) {
19         this.name = name;
20         this.topic = topic;
21         this.location = location;
22     }
23
24     public abstract String[] execute(TokenNode curCommand);
25     public abstract String[] execute(String curGesture);
26     public String getApplianceName() { return name; }
27     public void supportedGestures(String[] s) { this.supportedGestures.addAll(Arrays.asList(s)); }
28     void addExecutableWord(String[] s, String command) {
29         TokenNode root = new TokenNode(s[0]);
30         rootSet.add(root.getText());
31         root.setCommand(command);
32         if (s.length >= 2) {
33             TokenNode son = new TokenNode(s[1]);
34             root.addSon(son, index: 1);
35             if (s.length >= 3) {
36                 son.addSon(new TokenNode(s[2]), index: 1);
37             }
38         }
39         this.executableWords.add(root);
40     }
41     void addExecutableWords(String[][] s, String command) {
42         for (String[] each:s) {
43             addExecutableWord(each, command);
44         }
45     }
46     public Vector3D getLocation() { return location; }
47
48
49
50

```

Figure 29: The appliance class template

### 3.2.3.2 End-user Rule Template and Configuration File

Aside the user-initiated interaction to the appliance, the appliance also needs to be able to autonomously react when certain condition is met. This capability to allow the interaction initiated from both the user end and the appliance end is essential in most of the interactive architecture projects. For enabling the system triggered behaviors on the appliances, I decided to introduce the Rule concept into the backend system.

Rules are defined as the rule when certain conditions are met by the changes of user status, user movement, user activities and etc. Rules don't require the user to add any external sensors on the appliance but able to endow the appliance with awareness of the space and the user status.

Similar to the appliance configuration, configuring and applying rules on appliance is also very convenient by the help of the "rule.yml" file. (Figure 30) The user only needs to specify what rule he/she wants to add on which appliance, and the corresponding commands to

release when the condition is met. For different rules, the parameters might be different. My backend system provides four rules by default that can be applied to the appliance in the YAML file. The four rules are: Movement Rule, triggered if the user is moving; Special Zone Rule, triggered when the user enter a special zone in the space; Follow Rule, map the user's coordinates to the appliance; Proximity Rule; triggered when the user is within certain distance from the appliance. Table 1: Default Rule Parameter Table

*Table 1: Default Rule Parameter Table*

| <b>RULE TYPE</b> | Proximity.java                                  | SpecialZone.java                                                      | Movement.java                  | Follow.java                |
|------------------|-------------------------------------------------|-----------------------------------------------------------------------|--------------------------------|----------------------------|
| <b>PARAMETER</b> | distance<br>inRange Command<br>outRange Command | inRangeCommand<br>outRangdCommand<br>max X<br>max Y<br>min X<br>min Y | movingCommand<br>staticCommand | subscribingData<br>command |

If the user wants to create customized Rule, similar to the customizing appliance, the backend system exposes abstract rule class template (Figure 31) that the user can extend from and create their own condition. The users can write their logic in the “get Runnable” function and the backend system has a multithreaded mechanism for maintaining the execution of the rules.

```

1 plugins:
2   - name: soundModule
3     functions:
4       follow:
5         followingCommand: followingY
6
7   - name: LightSoundStrip
8     functions:
9       specialZone:
10        inRange: start
11        outRange: stop
12        maxX: "6000"
13        minX: "3000"
14        maxY: "7000"
15        minY: "0"
16
17   - name: lightStrip
18     functions:
19       follow:
20         followingCommand: "5"
21   - name: lightStrip
22     functions:
23       movement:
24         moving: "4"
25         stable: "0"
26
27   - name: light1
28     functions:
29       proximity:
30         distance: "3"
31         inRange: turn on
32         outRange: turn off
33

```

Figure 30: The rule applying file

```

1 package plugin;
2
3 import ...
4
5
6
7
8 public abstract class PluginTemplate {
9   protected MQTTSender mqtt;
10  protected MinuetSystem world;
11  protected Appliance targetObject;
12  protected int curStage = 0;
13  protected String name;
14  public String getUsername() {return userName; }
15  public void setUsername(String userName) { this.userName = userName; }
16  protected String userName;
17  protected boolean isRunning = true;
18  @ public PluginTemplate(Appliance targetObject) throws MqttException {
19    this.mqtt = new MQTTSender();
20    this.targetObject = targetObject;
21    this.name = targetObject.getApplianceName();
22    mqtt.sendMessage( topic: "connectedVoice", message: "$please select target object");
23  }
24
25  public String getName(){return this.name;}
26  public void kill(){ isRunning = false; }
27  public Appliance getTargetObject(){return this.targetObject;}
28  public void setWorld(MinuetSystem world) { this.world = world; }
29  abstract void getVoiceFeedback(String voice);
30  public abstract Runnable getTheRunnable();
31  protected enum Status{
32    On,Off
33  }
34
35  public void start() { world.runPlugin( pluginTemplate: this); }
36
37
38
39
40
41
42
43
44

```

Figure 31: The Rule class template

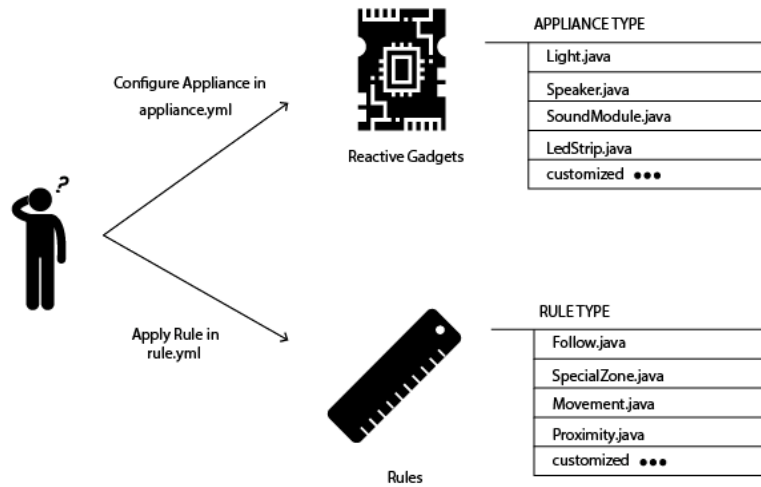


Figure 32: User development structure

### 3.3 Configuration Instruction

Registering appliance (Figure 32: User development structure) into the system requires the user to provide the name, the MQTT topic, the type, and the coordinates of the appliance. The name is a reference for identification purpose, which can be determined by the user and allow the system to search the corresponding appliance when the user wants to apply rules on it. The MQTT topic is the MQTT address, which is subscribed by the appliance, this address should be configure exactly the same as the appliance subscribing address. The type is for informing the system to instantiate the correct model in the backend system, the model is provided by default or can be extended from the existing template. The coordinate should be in the UWB RTLS coordinate system, which can be obtained by moving the end-user wristband close to the device and read from the MQTT topic.

Applying the rule (Figure 32: User development structure) to the appliance is asking the user to provide the type, the target appliance's name, and the parameters of the rule. The type of the rule can be referred from the default four rules or created be the end-user by following the template. The target name should be exactly the same as the appliance existing in the system, if

the system cannot find the appliance by the name, an exception would be thrown, and the user has to double check the settings. Each rule has its own set of parameters, the attribute names and the values should be filled in correctly.

## CHAPTER 4 EXAMPLE APPLICATION

### 4.1 Example Application

In this section, several interactive architecture projects were built as the example of using the system for indicating the efficiency, conveniences, scalability, intuitiveness that the system provides.

#### 4.1.1 Spatial Piano

Spatial Piano is an interactive architecture project which built upon the system introduced above. The goal of this project is for enabling the user's spatial experience with acoustic feedbacks. Transacting the movement in the space to become an experience of composition process. I customized a tiny Java app on the laptop to serve as the appliance in the system, which will play the different frequency of the keys of the piano. I applied the Follow Rule to the appliance, so that the app will follow the user's movement in Y direction of the space to play different audio files which contain the pre-recorded piano key sound. In this way, the system maps the user movement to the sound of piano. (Figure 34)

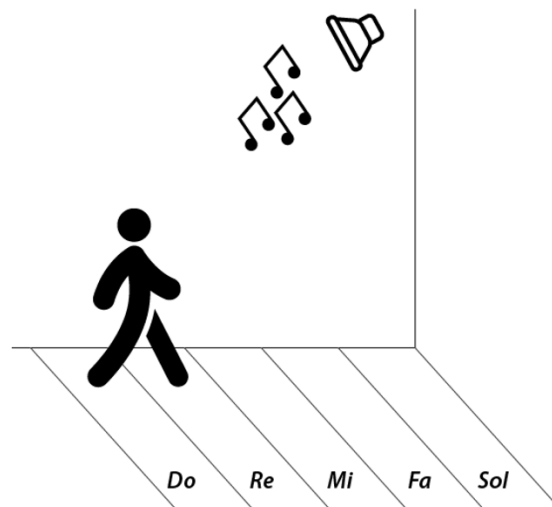


Figure 33: Spatial Piano

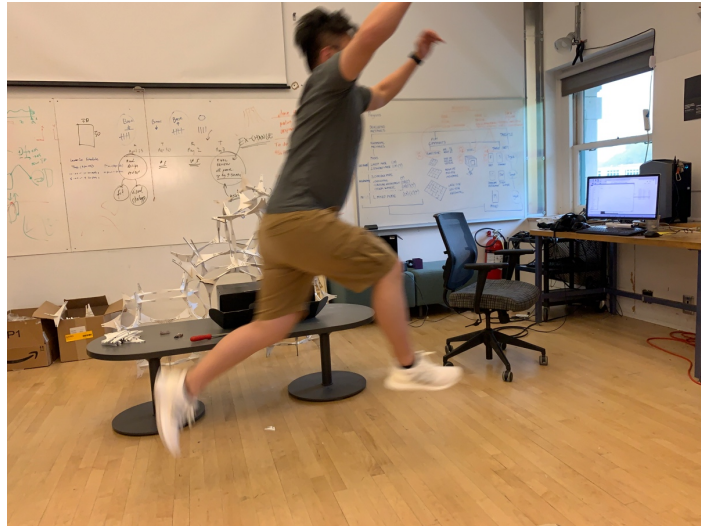


*Figure 34: The author was playing the Spatial Piano, which he wanted to trigger two un- contiguous keys by jumping*

#### **4.1.2 Voice Neon**

Voice Neon is another interactive project which was built by an LED light strip and a speaker. The end-user has to walk into a special area to trigger the speaker to listen to the voice of the user. The LED light strip will light up like a color neon by the frequency of the user's voice. The appliance was built by connecting an ESP8266 board and a WS2812 LED strip. The acoustic data is collected by the built-in mic of the laptop and conduct a Fast Fourier transform (FFT) to transform the sound signal to frequency domains, and then map to the lighting of different color and different positions of the LED light strip. The project was trying to create an unexpected experience which integrated the sound, the light, and the spatial experience of the user.

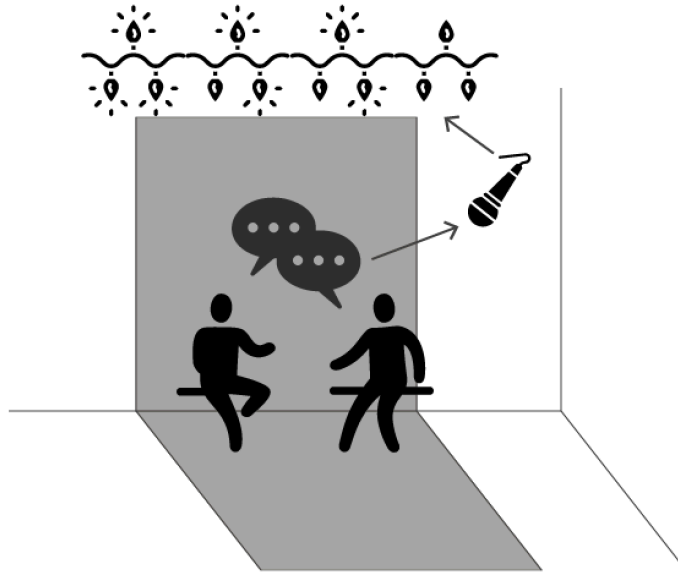


*Figure 35 Voice Neon*



*Figure 36: The author was experiencing the Voice Neon, the LED strip is reacting from his voice*

### **4.1.3 Smarter Home**

Unlike for the previous two projects that are more concentrated on the interactive and immersive art experience, the Smarter Home project is more focused on providing the practical

home automation experience for the occupants. The environment which has many commercial smart devices was augmented to become smarter. The smarter means the occupant doesn't need to use their mobile phone's apps to control the devices, nor do they need to use the intelligent voice assistants to help them release the commands. By simply pointing and using the gestures and verbal commands, they are able to inform the system with their selection and also release the commands. However, the project also considered the situation which is no easy to be achieved by existing solutions, such as providing more expressive instruction for the mobile vacuum to clean a specific spot. (Figure 38)



*Figure 37: Smarter Home*



*Figure 38: The author was controlling the Roomba by his gestures and voice*

## CHAPTER 5 HEURISTIC USER STUDY

For getting more insights from the end-user, a heuristic user study was conducted in CodeLab, Carnegie Mellon University. Four users (3 male, 1 female) participated in the study, all of them have different level of experience in architecture design, two of them had experiences in building interactive architecture projects, and all participants are resident in CodeLab.

In the study, I introduced and allowed them to set up the Spatial Piano, Voice Neon and Smarter Home projects by following the instruction. Due to the limitation of time, they were not required to build each project from sketch, but I provided them with the hardware appliances that already had working firmware been flashed. Each person was required to configure the appliance for the system and apply rules to control the device. After configuration and applying rules, they were allowed to experience the project they just built. I recorded their reaction and feedback of the system.

Overall, the participants welcomed the idea of introducing voice, gestures and spatial information interaction in the interactive architecture project. *“I like the idea of supporting both voice and gesture, which is similar to how people interact with each other every day.”* (P3) Moreover, by allowing the end-user to use file-based method to configure appliance also raised the user’s interest. *“I think it is a good idea to enable the device registration, it will be better if I have an interface to drag and drop.”* (P1) What’s more, the scalability and extensibility also received positive feedbacks. *“In this Object-oriented way of system implantation, I only need to follow the template for customized my own appliance and rules which are compatible with the system.”* (P4) Last but not the least, most of the participants agreed that this system not only extending the sensing capability of the interactive architecture but also provides a boosting solution for building interactive architecture project. (P2, P3, P4) *“This system is very interesting*

that it serves as a platform for us to create interactive projects, but also frees me from implementing those data gathering and parsing processes.” (P1)

However, the participants also provided valued feedbacks about the limitation of the system. 1. Two of the users experienced the situation that the voice recognition system could not successfully translate their voice commands, which might cause from their accent and the environmental noise. 2. Also, the system instruction is still complicated for the user who doesn't have any experience on implementing interactive project. Another layer of abstraction of the system instruction will be necessary for non-expert. 3. What's more, the amount of default appliances and rules is not enough. Due to the fact that users would tend to use the existing examples for understanding the structure and instruction of the system, more well-documented examples and default appliances are required for helping them to implement their own projects.

Except for the observed limitations, the objective limitations are raised as follow: 1. The setup process for the system's UWB anchors and the MQTT broker is still required, which involved more labor than computer vision-based implementation. 2. The UWB-based real-time localization system depends on Light-of-Sight context, which will be hard to ensure in a complex indoor environment. 3. The latency of the cloud-based voice tracking system is hard to decrease under the limitation of net speed.

Eliciting from the observed limitation and objective limitation, I discovered (Figure 39) that on average participants' pointing was off about  $9.0^\circ$  ( $SD=4.7^\circ$ ) and collectively 95% of the pointing trials fell within a margin of error of  $17.7^\circ$ . These findings inform the choice of pointing angle triggering threshold.

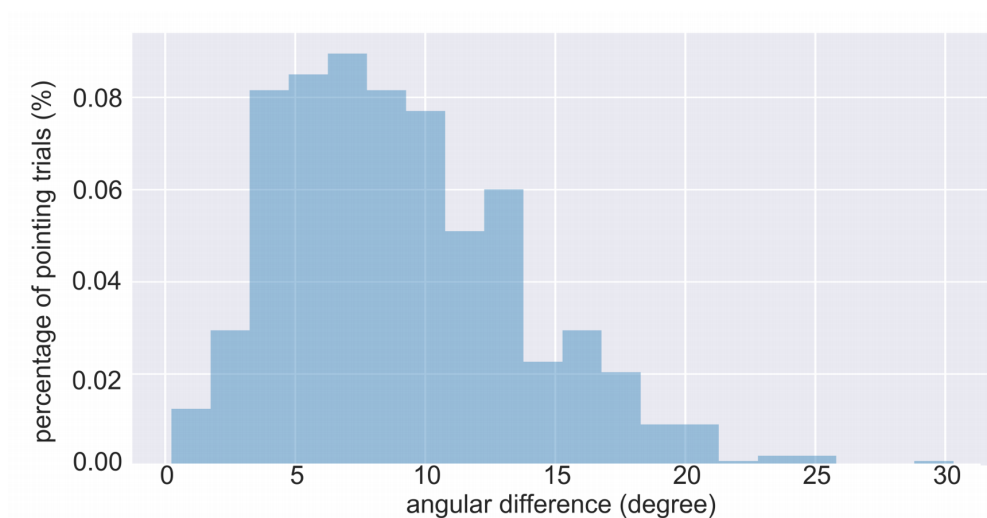


*Figure 39: The distribution of angular difference*

## CONCLUSION and FUTUREWORKS

In this research, aiming for researching *How to enable rapid prototyping design processes for creating extensible and efficient interactive spaces which also provides intuitive interaction and closer relationships between the occupant, device, and the environment*, I presented an sensing platform which allows the designers to conveniently create, configure and interact with smart appliances with gesture, voice and spatial movement. Besides normal user-initiated interaction, I provided another Rule based interaction which is reactive to the user movement, status, and activities. I also provide tools for easing the process of appliance configuration, rule application and gesture training. All of these parts in together create a design pipeline and design tools for designers to create interactive appliances in their interactive architecture projects.

The system still has following aspects that need to be improved in the future. First of all, the robustness of the recognition system. Due to the fact that this is a research which is more focus on the proof-of-concept aspect, the gestural recognition and the voice recognition parts haven't been investigated too deep for improving the performance. For the voice recognition part, the system has hard time recognizing the voice if the user has accent, which can be solved by turning and adjusting the voice model corresponding to each user. The gestural recognition part can be further improved by comparing with different machine learning models. Secondly, the system still requires the end user to wear a sensing wristband and deploy four UWB anchors in the space, also the power consumption of the UWB sensor also need to take into consideration for future research. Last but not the least, providing a graphical user interface for device configuration and rule applying, as for now I provided a file-based solution for configuration

purpose, which can be replaced by a customized user interface which will also provide in place instructions and error checking.

## References

Bolt, Richard A. "'Put-that-there'." Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '80, 1980.

doi:10.1145/800250.807503.

Bullivant, Lucy. *4dspace: Interactive Architecture*. Wiley-Academy, 2005.

Cvijikj, Irena Pletikosa, and Florian Michahelles. "The Toolkit Approach for End-user Participation in the Internet of Things." *Architecting the Internet of Things*, 2011, 65-96.

doi:10.1007/978-3-642-19157-2\_4.

Dourish, Paul. *Where the Action Is the Foundations of Embodied Interaction*. Bradford Books, 2004.

Dacosta, Francis. "Architecture for the Frontier." *Rethinking the Internet of Things*, 2013, 95-121. doi:10.1007/978-1-4302-5741-7\_6.

Fox, Michael. *Interactive Architecture Adaptive World*. Princeton Architectural Press, 2016.

Frazer, John H. "The Architectural Relevance of Cybernetics." *Systems Research* 10, no. 3 (01, 2007): 43-48. doi:10.1002/sres.3850100307.

Frazer, John. *An Evolutionary Architecture*. Architectural Association, 1995.

"Gauss-Weber Telegraph – 1833." MagLab.<https://nationalmaglab.org/education/magnet-academy/history-of-electricity-magnetism/museum/gauss-weber-telegraph>.

Ghajargar, Maliheh, Mikael Wiberg, and Erik Stolterman. "Designing Places for Reflection." *Internet of Things Social Internet of Things*, 07, 2018, 107-20. doi:10.1007/978-3-319-94659-7\_6.

Heyl, Parker. "Analog Future." Interactive Architecture Lab. January 07, 2019.

<http://www.interactivearchitecture.org/lab-projects/analog-future>.

Hutchins, Edwin. *Cognition in the Wild*. MIT Press, 2006.

Jenson, Scott. "The Future IoT: Building Better Legos." *Computer* 50, no. 2 (02 2017): 68-71. doi:10.1109/mc.2017.48.

Jimenez, A.r., and F. Seco. "Comparing Decawave and Besspoon UWB Location Systems: Indoor/outdoor Performance Analysis." *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 10 2016. doi:10.1109/ipin.2016.7743686.

Journal, RFID. "That 'Internet of Things' Thing." That 'Internet of Things' Thing - 2009-06-22 - Page 1 - RFID Journal. <https://www.rfidjournal.com/articles/view?4986>.

Laput, Gierad, Robert Xiao, and Chris Harrison. "ViBand." *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16*, 2016. doi:10.1145/2984511.2984582.

Mansell, Robin. *The Oxford Handbook of Information and Communication Technologies*. Oxford University Press, 2007.

Marquardt, Nicolai. "Proxemic Interactions in Ubiquitous Computing Ecologies." *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '11*, 2011. doi:10.1145/1979742.1979691.

Matalucci, Berardo, Kenton Phillips, Alicia A. Walf, Anna Dyson, and Joshua Draper. "An Experimental Design Framework for the Personalization of Indoor Microclimates through Feedback Loops between Responsive Thermal Systems and Occupant Biometrics." *International Journal of Architectural Computing* 15, no. 1 (03 2017): 54-69. doi:10.1177/1478077117691601.

Mayton, Brian D., Nan Zhao, Matt Aldrich, Nicholas Gillian, and Joseph A. Paradiso.

"WristQue: A Personal Sensor Wristband." *2013 IEEE International Conference on Body Sensor Networks*, 05 2013. doi:10.1109/bsn.2013.6575483.

Mazzei, Daniele, Gualtiero Fantoni, Gabriele Montelisciani, and Giacomo Baldi.

"Internet of Things for Designing Smart Objects." *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 03 2014. doi:10.1109/wf-iot.2014.6803175.

Oviatt, Sharon, Phil Cohen, Lizhong Wu, Lisbeth Duncan, Bernhard Suhm, Josh Bers, Thomas Holzman, Terry Winograd, James Landay, Jim Larson, and David Ferro. "Designing the User Interface for Multimodal Speech and Pen-Based Gesture Applications: State-of-the-Art Systems and Future Research Directions." *Human-Computer Interaction* 15, no. 4 (12 2000): 263-322. doi:10.1207/s15327051hci1504\_1.

Oviatt, Sharon, Antonella Deangeli, and Karen Kuhn. "Integration and Synchronization of Input Modes during Multimodal Human-computer Interaction." *Referring Phenomena in a Multimedia Context and Their Computational Treatment on - ReferringPhenomena '97*, 1997. doi:10.3115/1621585.1621587.

Oviatt, Sharon. "Ten Myths of Multimodal Interaction." *Communications of the ACM* 42, no. 11 (11, 1999): 74-81. doi:10.1145/319382.319398.

"Protocols of the Wireless Internet of Things." *The Wireless Internet of Things*, 10, 2018, 21-45. doi: 10.1002/9781119260608.ch2.

"RED." Node. <https://nodered.org/>.

Rose, David. *Enchanted Objects: Design, Human Desire, and the Internet of Things*. Scribner, 2014.

Schilit, B., N. Adams, and R. Want. "Context-aware Computing Applications." *Workshop on Mobile Computing Systems and Applications*. doi:10.1109/mcsa.1994.512740.

Schoenberger, Chana R. "The Internet of Things." *Forbes*. June 19, 2013.  
<https://www.forbes.com/global/2002/0318/092.html#6d3e3f873c3e>.

Stivers, Tanya, and Jack Sidnell. "Introduction: Multimodal Interaction." *Semiotica* 2005, no. 156 (01, 2005): 1-20. doi:10.1515/semi.2005.2005.156.1.

Sun, Yong, Fang Chen, Yu (David) Shi, and Vera Chung. "A Novel Method for Multi-sensory Data Fusion in Multimodal Human Computer Interaction." *Proceedings of the 20th Conference of the Computer-human Interaction Special Interest Group (CHISIG) of Australia on Computer-human Interaction: Design: Activities, Artefacts and Environments - OZCHI '06*, 2006. doi:10.1145/1228175.1228257.

Susi, Melania, Valérie Renaudin, and Gérard Lachapelle. "Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users." *Sensors* 13, no. 2 (01, 2013): 1539-562. doi:10.3390/s130201539.

Tachi, Tomohiro. "Interactive Freeform Design of Tensegrity." *Advances in Architectural Geometry* 2012, 2013, 259-68. doi:10.1007/978-3-7091-1251-9\_21.

Ushigome, Yosuke, Ryuma Niiyama, Kunihiro Nishimura, Tomohiro Tanikawa, and Michitaka Hirose. "Archi/e Machina: Interactive Architecture Based on Tensegrity." *2010 16th International Conference on Virtual Systems and Multimedia*, 10 2010. doi:10.1109/vsimm.2010.5665965.

Vermesan, Ovidiu, and Peter Friess. "Digitising the Industry - Internet of Things Connecting the Physical, Digital and Virtual Worlds." *Digitising the Industry - Internet of Things*

*Connecting the Physical, Digital and Virtual Worlds*, 2016, 1-364. doi:10.13052/rp-9788793379824.

Weiser, Mark. "The Computer for the 21st Century." *ACM SIGMOBILE Mobile Computing and Communications Review* 3, no. 3 (07, 1999): 3-11. doi:10.1145/329124.329126.

Xiao, Robert, Gierad Laput, Yang Zhang, and Chris Harrison. "Deus EM Machina." *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, 2017. doi:10.1145/3025453.3025828.

## Appendix

**Appliance Template Java Code:**

```
package model;

import javafx.util.Pair;

import org.apache.commons.math3.geometry.euclidean.threed.Vector3D;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

public abstract class Appliance {

    private String name;

    private Vector3D location;

    private String topic;

    protected Set<TokenNode> executableWords = new HashSet<>();

    protected Set<String> supportedGestures = new HashSet<>();

    protected Set<String> rootSet = new HashSet<>();

    public Appliance(String name, Vector3D location,String topic) {

        this.name = name;

        this.topic = topic;

        this.location = location;
```

```

}

public abstract String[] execute(TokenNode curCommand);

public abstract String[] execute(String curGesture);

public String getApplianceName() {
    return name;
}

public void supportedGestures(String[] s) {
    this.supportedGestures.addAll(Arrays.asList(s));
}

void addExecutableWord(String[] s,String command) {
    TokenNode root = new TokenNode(s[0]);
    rootSet.add(root.getText());
    root.setCommand(command);
    if (s.length>=2) {
        TokenNode son = new TokenNode(s[1]);
        root.addSon(son,1);
        if(s.length >= 3){
            son.addSon(new TokenNode(s[2]),1);
        }
    }
    this.executableWords.add(root);
}

void addExecutableWords(String[][]s,String command){

```

```

    for (String[] each:s) {
        addExecutableWord(each,command);
    }
}

public Vector3D getLocation() {
    return location;
}

public boolean checkBePointed(Pair<Vector3D, Vector3D> userLocAndPointVec) {
    Vector3D target = userLocAndPointVec.getKey();
    Vector3D pointingVec = userLocAndPointVec.getValue();
    Vector3D oc = location.subtract(target);
    double projectoc = oc.dotProduct(pointingVec);
    if (projectoc<=0)return false;
    double oc2 = oc.dotProduct(oc);
    double distant2 = oc2 - projectoc*projectoc;
    if (Math.toDegrees(Vector3D.angle(oc, pointingVec)) <= 20) {
        System.out.println("angle= "+ Math.toDegrees(Vector3D.angle(oc,pointingVec)));
        System.out.println("distant= "+ Math.sqrt(distant2));
    }
    return (Math.toDegrees(Vector3D.angle(oc,pointingVec))<=20.0);
}

public boolean canExeVoiceCommand(TokenNode userCommand) {

```

```

for(TokenNode command : this.executableWords){
    if(NLPHandler.isExecutable(command,userCommand))return true;
}
return false;
}

```

```

public boolean canExcuGestureCommand(String gesture){
    if(supportedGestures.contains(gesture)){
        return true;
    }
    return false;
}

```

```

public String getTopic() {
    return topic;
}
}

```

**Rule Template Java Code:**

```
package plugin;

import model.Appliance;

import model.MQTTSender;

import model.MinuetSystem;

import org.eclipse.paho.client.mqttv3.MqttException;


public abstract class PluginTemplate {

    protected MQTTSender mqtt;

    protected MinuetSystem world;

    protected Appliance targetObject;

    protected int curStage = 0;

    protected String name;

    public String getUserName() {return userName; }

    public void setUserName(String userName) {

        this.userName = userName;

    }

    protected String userName;

    protected boolean isRunning = true;

    public PluginTemplate(Appliance targetObject) throws MqttException {

        this.mqtt = new MQTTSender();

        this.targetObject = targetObject;

        this.name = targetObject.getApplianceName();

    }

}
```

```

        mqtt.sendMessage("connectedVoice","$please select target object");
    }

    public String getName(){return this.name;}

    public void kill() {
        isRunning = false;
    }

    public Appliance getTargetObject(){return this.targetObject;}

    public void setWorld(MinuetSystem world) {
        this.world = world;
    }

    abstract void getVoiceFeedback(String voice);

    public abstract Runnable getTheRunnable();

    protected enum Status{

        On,Off

    }

    public void start(){
        world.runPlugin(this);
    }

}

```