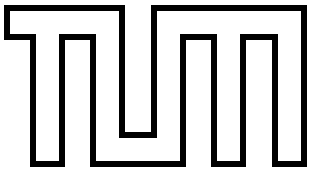
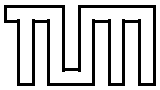


FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

MIBO – A Framework for the Integration of Multimodal Intuitive Controls in Smart Buildings

Sebastian Matthias Peters



FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Forschungs- und Lehreinheit 1
Angewandte Softwaretechnik

MIBO – A Framework for the Integration of Multimodal Intuitive Controls in Smart Buildings

Sebastian Matthias Peters

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr. Hans-Michael Gerndt
Prüfer der Dissertation:	1. Univ.-Prof. Bernd Brügge, Ph.D.
	2. Prof. Vivian Loftness, Ph.D. Carnegie Mellon University, Pittsburgh, PA, U.S.A.

Die Dissertation wurde am 10.05.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 04.07.2016 angenommen.

Acknowledgments

This work would not have been possible without the support of so many people. I would like to thank everybody who supported me during my research and thus contributed to it. It is not even possible to name everybody on a single page but please be sure that I am very thankful for everyone's help.

First, I would like to express my deep gratitude to Bernd Brügge and Vivian Loftness, who have been much more to me than supervisors. Your infinite inspiration, passion and enthusiasm has clearly driven my motivation in this research during these four years. With your trust in me and my work, you gave me the freedom to run my own research as an independent and responsible scientist. You have both created invaluable environments of opportunities and creativity in Munich and Pittsburgh – thank you for everything!

I would also like to thank all members of the Chair for Applied Software Engineering. I learned a lot from all of you, and I am indebted for your support and the social environment you have all contributed to. Special thanks to Stephan Krusche for your helpful advices and reviews. At the same time, I am very grateful to all members of the Intelligent Workplace at Carnegie Mellon University for supporting me during my research abroad and for keeping up the friendship and the invaluable exchange between our institutions.

I thank my students and co-authors who contributed to my research, in particular Arno Schneider, Jan Ole Johanßen, Dominic Henze, Stefan Nosovic, Nadine v. Frankenberg and Masashi Beheim.

Finally, I would like to express my love and gratitude to my family, and to my girlfriend Mira. Writing a dissertation requires more than a researcher's full attention. I am indebted for your love and devotion, your constant support and understanding. You are the source of joy and happiness in my life.

Abstract

With the *Internet of Things* expanding into our homes and offices, we can expect fixtures in buildings to become more interconnected and more numerous as part of an increasingly complex and powerful integrated smart environment. However, this raises new challenges in the area of usability since today's rooms are already cluttered with multiple user interfaces in the form of buttons and remote controls. In the future, as more technology is embedded into buildings, users must have the opportunity to choose and to combine a heterogeneous set of devices and modalities. *Multimodal User Interaction* technology aims at creating natural and intuitively usable interfaces, allowing a user to interact with systems in a way similar to human-to-human communication. The combination of modalities, such as gestures, speech and gaze-tracking provides occupants with an integrated and intuitive interface for diverse addressable fixtures in buildings. However, creating multimodal interfaces remains a complex and highly specialized task.

This research describes *MIBO* – a framework to enable and define multimodal intuitive building controls. It supports the integration and combination of multiple interaction modalities, e.g., gesture recognition, speech recognition, eye-tracking but also traditional button controls and performs the multimodal fusion. It also provides a meta model and an extensible domain-specific language to describe a multimodal interaction model and to separate it from its implementation. This enables prototyping of new control systems for developers and allows end-users the configuration of desired interactions in a building. The framework has been applied in three real-world applications, each in at least two different buildings with more than 20 users, demonstrating the feasibility and efficiency of our approach.

Contents

Acknowledgements **v**

Abstract **vii**

1. Introduction **1**
1.1. Problem Statement 2
1.2. Contribution 3
1.3. Scope 4
1.4. Research Approach 5
1.5. Outline of the Dissertation 5

2. Foundations **7**
2.1. Ubiquitous Computing 8
2.2. Ambient Intelligence 8
2.3. Smart Environments 9
2.3.1. Smart Buildings 11
2.3.2. Context Awareness 16
2.3.3. Energy Awareness 18
2.3.4. Disability Awareness 19
2.4. Cyber-Physical Systems 20
2.5. Internet of Things 21
2.6. Natural User Interfaces 22
2.6.1. Unimodal User Interfaces 22
2.6.2. Multimodal User Interfaces 26
2.7. Intuitive User Interfaces 29

3. Requirements for Intuitive Controls in Smart Buildings **33**
3.1. Methodology 33

3.2. Requirements Identification . . . 35
3.2.1. Study Design . . . 36
3.2.2. Results . . . 38
3.3. Visionary Scenarios . . . 40
3.3.1. Scenario 1: Intuitive Building Control . . . 41
3.3.2. Scenario 2: Building Configuration . . . 41
3.3.3. Scenario 3: Prototyping . . . 41
3.3.4. Scenario 4: Ambient Assisted Living . . . 42
3.4. Application Domain Requirements . . . 42
3.5. Solution Domain Requirements . . . 47
3.6. Summary . . . 47

4. Framework Design **49**
4.1. Introduction . . . 49
4.1.1. Multimodal Parsing and Integration . . . 49
4.1.2. Architectures and Frameworks for Multimodal Interfaces . . . 51
4.1.3. Building Automation and Control Systems . . . 54
4.2. Application Domain . . . 57
4.3. MIBO Software Architecture . . . 61
4.3.1. Blackboard Architectural Style . . . 62
4.3.2. Subsystem Decomposition . . . 65
4.3.3. Multimodal Fusion and Integration . . . 68
4.3.4. Multi-Device User Feedback . . . 69
4.3.5. Access Control and Security . . . 70
4.3.6. Design Rationale . . . 71

5. MiboML – A DSL for Multimodal Interaction Models in Smart Buildings **73**
5.1. Design Goals of MiboML . . . 74
5.2. MiboML Grammar . . . 76
5.2.1. Core Grammar . . . 76
5.2.2. Modality Extension Grammar . . . 77
5.2.3. Grammar Parsing . . . 77
5.3. Lexical Representation . . . 78
5.4. Conflict Resolution . . . 80
5.5. IDE for Multimodal Interaction Models . . . 82

6. Case Studies **85**
6.1. HomeGestures – A Gesture-based Smartphone Control for Smart Buildings 85
6.1.1. Interaction Design . . . 86
6.1.2. Implementation . . . 87
6.1.3. Related Work . . . 90
6.1.4. Evaluation . . . 93
6.2. NICE – Hands-free Natural User Interfaces in Smart Buildings . . . 94
6.2.1. Interaction Design . . . 95
6.2.2. Implementation . . . 96
6.2.3. Related Work . . . 97
6.2.4. Evaluation . . . 99
6.3. SISSI – Smart Interface for Speech Service Integration . . . 104
6.3.1. Interaction Design . . . 105
6.3.2. Implementation . . . 107
6.3.3. Related Work . . . 110
6.3.4. Evaluation . . . 112

7. Conclusion **119**
7.1. Contributions . . . 120
7.2. Limitations . . . 123
7.3. Future Work . . . 124

Appendix **131**

A. Glossary **131**

B. History of Gesture-based User Interfaces **133**

C. Voice-based User Interfaces for Smart Buildings **135**

Bibliography **137**

1. Introduction

Modern sedentary lifestyles have led to the fact that a majority of Americans spend approximately 22 hours per day indoors [BLS14]. Within buildings, lighting comfort, thermal comfort and indoor air quality rank among the top ten leading tenant complaints [IFMA12]. These conditions and their effects on occupants and residents are measured by *Indoor Environmental Quality (IEQ)*. Better indoor environmental quality can enhance the lives of building occupants and increase the resale value of the building [Coyle14]. Particularly, in office buildings, increasing the IEQ level and thus, the employees' health and productivity over the long run, can have a large return on investment, since personnel costs of salaries typically surpass building operating costs.

Hartkopf, Loftness et al. have shown that occupant behavior and feedback are necessary to achieve the desired building performance [Hartkopf86] and research by Park, Choi and Daum has proven that occupants can be a good sensor and controller for IEQ performance [Park15] [Choi10] [Daum11]. Furthermore, Gu has shown that the ability to individually adjust heating, lighting and ventilation influences the IEQ and thus well-being, satisfaction and productivity of occupants [Gu11].

Additionally, the aspect of energy use in buildings is intrinsically tied to IEQ performance. Gu has shown that providing the users with individual control can have a positive impact on the energy consumption of occupants [Gu11]. This is remarkable when taking into account that buildings in the United States account for 41% of total energy use and 73% of total electricity consumption [DOE13]. Therefore, energy savings in buildings are crucial to reducing our dependence on fossil fuels and greenhouse gas emissions.

This research assumes that providing individual control to building occupants is a key to comfort, well being and energy efficiency. However, this control must be simple, fast and natural.

1.1. Problem Statement

Buildings tend to rely on large zones of control [Loftness09], meaning that lighting, heating and ventilation are controlled for large areas with many occupants at once. This results in both, discomfort for occupants and a waste of energy because each occupant may have different needs, depending on the individual task, clothing or habits [Park15]. Large control zones potentially tend to over-cool, over-heat and over-light entire areas, that may only be occupied by few people.

On the other hand, we are currently observing the *Internet of Things* (see section 2.5) expanding into buildings, making every fixture addressable over the internet. Furthermore, we can expect these fixtures to become more interconnected, more numerous and more specialized as part of an increasingly complex and powerful integrated intelligent environment [Wilson03]. However, this raises new challenges in the area of usability. Today's rooms are already cluttered with multiple user interfaces in the form of buttons and remote controls, each covered with even more buttons [Wilson03]. These interfaces require the user to devote attention to finding the right button rather than paying attention to the environment. As new technologies are released and more functionality is continuously added to buildings, they will become more complex to use if we hold on to these traditional controls. This is important in particular considering the trend that people tend to avoid reading user manuals [Novick06].

We consider Mark Weiser's vision of *Ubiquitous Computing* (see section 2.1) as a solution to these usability problems [Weiser91]. Weiser predicted that computing devices will become invisible and that traditional desktop-oriented paradigms will be replaced by more sophisticated devices working in the background. These devices *weave themselves into the fabric of everyday life until they are indistinguishable from it* [Weiser91]. With the current technologies at hand, we may partially realize Weiser's vision, although there are still research issues to be solved to make significant progress. This dissertation aims at addressing several of these related issues.

As future interaction is embedded into our buildings, developers have to consider usability beyond the traditional WIMP (windows, icons, menus, pointers) paradigm. *Multimodal User Interface (MMUI)* technology aims at providing natural and intuitive interfaces, allowing a user to interact with systems in a way similar to human-to-human communication [Sun06]. Modalities, such as gestures, speech and gaze-tracking may be combined flexibly to form an effective and intuitive control set for the occupants of a building. Eventually, the synergistic connection between multimodality and pervasivity in our future

buildings allows breaking down barriers in human-computer interaction and make communication spontaneous [D'Andrea09].

The research community agrees that creating multimodal interfaces remains a complex and highly specialized task [Johnston09a]. Therefore, we propose tailored solutions for complex domains, such as smart buildings, because the interactions in a building can be quite different to other multimodal controls, such as multitouch gestures on a screen, pen input and others. Additionally, as the range of multimodal input is extended, it becomes essential to provide a declarative statement of the grammar of multimodal interaction and separate from the parsing algorithms. This enables system developers to describe integration strategies in a high level representation, facilitate prototyping and iterative development of multimodal systems [Johnston98b].

1.2. Contribution

This dissertation presents *MIBO* – a framework for the definition and integration of multimodal intuitive controls in smart buildings. The framework eases the development of multimodal building controls in several dimensions. It supports the integration of multimodal interactions, e.g. using gesture recognition, speech recognition, eye-tracking. It defines a meta model and an extensible domain-specific language to describe multimodal interaction models for smart buildings. An interaction model describes the particular interactions that a user can perform to issue a control command for a fixture. It consists of many definitions, each describing one way to issue a control command for a defined scope of fixtures. This separates the definition of the interactions and the multimodal fusion from its implementation and allows for prototyping of new control systems by developers. Eventually, it allows facility managers and end users to define and adapt the interactions in a building. Figure 1.1 shows the context of MIBO in a class diagram.

The framework provides a reference architecture, consisting of a blackboard for the resolution of the multimodal input commands and the representation of knowledge as self-activating, asynchronous, parallel processes. It provides an extensible architecture for multimodal controls in smart buildings. The use of different controls in multiple buildings has been shown in several case studies (see chapter 6). These studies show the feasibility and efficiency of our approach in real-world applications, each in at least two different buildings with different users.

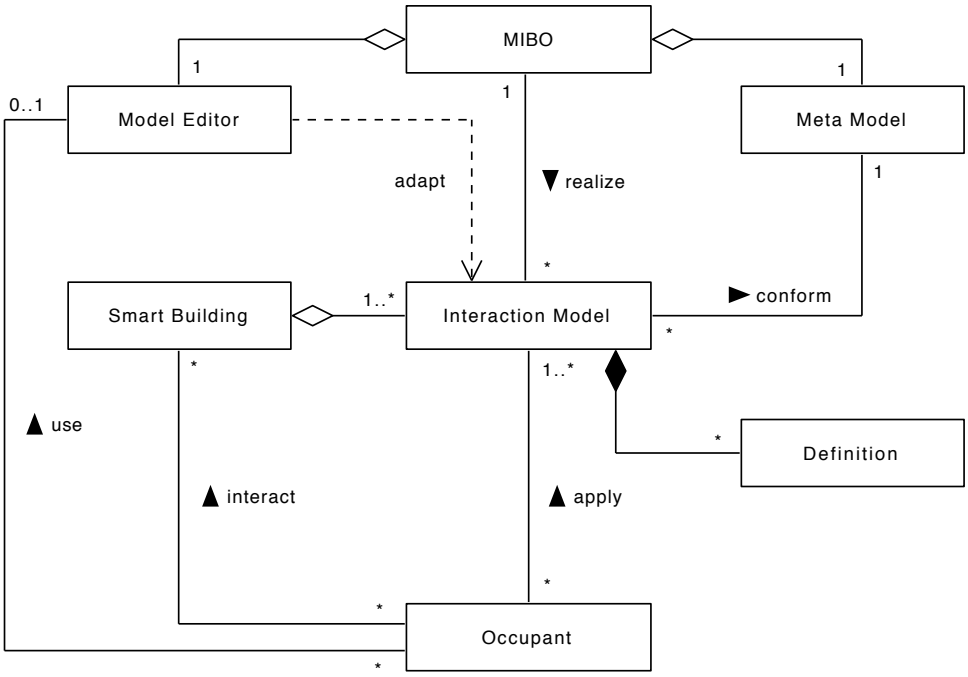


Figure 1.1.: Context of MIBO as UML class diagram.

1.3. Scope

The scope of the dissertation is to provide a meta model and a reference architecture for the multimodal integration of controls in smart buildings and demonstrate its feasibility in real-world applications. The goal is to control fixtures in a smart building using natural and intuitive interfaces allowing the occupants to interact with their environment in a way similar to human-to-human communication. This should lower the access barriers and keep the human in the loop to improve lighting comfort, thermal comfort and indoor air quality, based on the individual needs. Finally, user studies show the effectiveness of multimodal controls in smart buildings. The focus is on the aspects of usability, extensibility and scalability of the framework. In order to determine suitable modalities and interactions to control buildings, we also present a hybrid methodology combining two bottom-up methodologies with a top-down approach to provide a user-centered, participatory design while ensuring technical factors that are not covered by user experiments.

Privacy and Security are important criteria of smart buildings, especially when taking into account the *context* of the user and the environment. However, privacy and security are not in the focus of this dissertation.

1.4. Research Approach

This research is guided by narrative descriptions of envisioned usage, called scenarios. Scenario-based design is a family of techniques in which the use of a future system is concretely described at an early point in the development process [Rosson02]. Since the use experience and the user are considered to be the guiding factors of the development, this methodology is referred to as a user-centered approach [ISO 9241-210:2010].

Multimodal intuitive interaction requires a methodology that embraces both, usability engineering and software engineering. Göransson et al. [Goransson03] showed the extension of the Unified Process (UP) with an additional task called *Usability Design*. The idea is to include usability as a major focus from the very beginning of the project without having an existing system. As this is a user-centered project, the usability process is embedded throughout the whole project phase.

We started our research with a wizard-of-oz usability experiment to investigate the varying user behavior in an imaginary smart environment with ubiquitous computing technology by anecdotal evidence. Based on this, the requirements for a reusable and flexible framework were identified. Visionary scenarios guided the formulation of the requirements. Afterwards, a reference architecture has been designed and implemented. It has been tested in three case studies for different controls, each employed in at least two different buildings. Formative evaluations of the case studies have been conducted to validate our approach of multimodal controls in smart buildings.

1.5. Outline of the Dissertation

Chapter 1 introduces the notion of indoor environmental quality (IEQ), energy efficiency in buildings and derives challenges for intuitively usable building controls. Next, the scope and approach of this research is defined.

Chapter 2 introduces the foundations of Ubiquitous Computing, Ambient Intelligence, Smart Environments, Natural User Interfaces, Multimodal Interaction, the Internet of Things and others.

Chapter 3 elicits the requirements for intuitive building controls by combining a guessability study and wizard-of-oz experiment to learn about the requirements for intuitive multimodal controls in buildings. Based on these results, visionary scenarios and func-

tional requirements define the scope of MIBO. Finally, non-functional requirements are described.

Chapter 4 describes the design of the MIBO framework for multimodal intuitive building controls. It starts with a comprehensive review of related work for multimodal integration and frameworks for multimodal interfaces. Afterwards, the application domain model and the reference architecture of MIBO is presented.

Chapter 5 describes the extensible multimodal grammar *MiboML* which allows the declarative definition of multimodal interaction in smart buildings. This allows for the separation of interaction models from the actual implementation.

Chapter 6 presents three case studies within which we tested and leveraged the framework. A gesture-based smartphone control, a hands-free natural user interface and a voice-based smartphone controller for smart buildings are presented and evaluated.

Chapter 7 concludes with an overview of the contributions and limitations of the results and sketches future work.

2. Foundations

We consider Mark Weiser's vision of *Ubiquitous Computing* as the embracing concept for smart environments and this research. Weiser describes an environment where computing devices become less visible, up to the point where users become unaware of the fact that they are using computing systems (see section 2.1). Over the past decades, ubiquitous computing has been augmented with additional trending terms, such as Smart Environments, Ambient Intelligence, Ambient Assisted Living, Internet of Things, Cyber-Physical Systems and others. Additionally, in the domain of human-computer interaction, terms like Natural User Interfaces, Multimodal User Interfaces, User Experience and Intuitive Use have emerged. They are all used widely in today's research as well as in commercial products. This chapter aims at explaining the relation of each topic to this dissertation and bringing the terms into a meaningful structure.

While Weiser stressed his idea of invisibility and unobtrusiveness, *Ambient Intelligence* (see section 2.2) describes the vision of adding artificial intelligence to ubiquitous computing. Furthermore, a *Smart Environment* (see section 2.3) is the representation of a physical space, which combines perceptual and reasoning capabilities with the other elements of ubiquitous computing. We consider smart environments to be a superclass of diverse physical spaces, such as smart buildings, smart cars, smart airplanes or even smart cities. A smart environment does not only provide some service automation, but learns and adapts its behavior during use, taking into account the context of the environment as well as the context of the user. The user is always put in the center of all decision processes and retains control over the environment using natural and intuitive controls. In contrast to traditional graphical user interfaces (GUIs), natural user interfaces (see section 2.6) are invisible to the user, for instance applying voice, gestures and gaze-tracking. Due to the fact that they are invisible, they can be used to realize ubiquitous computing. The combination of several of these modalities (e.g. speech and gestures) is called multimodal user interaction (see section 2.6.2). An interface which is usable *intuitively* refers to the aspect of familiarity and learnability (see section 2.7).

2.1. Ubiquitous Computing

Mark Weiser formulated his vision of *Ubiquitous Computing* in the early 90's in *"The computer for the 21st Century"* [Weiser91].

> "Specialized elements of hardware and software, connected by wires, radio waves and infrared, will be so ubiquitous that no one will notice their presence. The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [Weiser91]

Weiser claimed that the traditional desktop-oriented paradigms will be replaced by more sophisticated devices working in the background and thereby, computers would become less visible and better suit the human environment. Toth et al. argue that the main goal of ubiquitous computing systems is to offer a user interface so natural and easy, that users become unaware of the fact they are using a computing system [Tóth09]. Ubiquitous computing also embraces the notions of *Pervasive Computing* and *Mobile Computing*. Pervasive computing is the combination of distributed computing with sensors and actors in an environment. Mobile computing enables users to include a personalized device which is not bound to a specific location and which integrates the personal context of the user, such as current location, movement, calendar, contacts and others.

Since the formulation of Weiser's vision, computing devices have shrunk dramatically from big mainframes to microchips that can be embedded in a variety of places. Today, some appliances in buildings have become integrated to such an extent that we use them without consciously thinking about them [Augusto10b]. With the increasing pervasion of technology in our daily life, and at the same time, decreasing awareness, challenges about usability are emerging. Even computing devices in our houses become fully invisible, which requires approaches to retain control about them. Our hypothesis is that multimodal interaction provides considerable benefits for users to control their environment in an intuitive and unobtrusive way.

2.2. Ambient Intelligence

The necessary coordination of distributed computing devices by intelligent systems has led to the introduction of *Ambient Intelligence (AmI)*, which is a confluence of ubiquitous

computing and artificial intelligence. It started to be used as a term to describe this development at the end of the nineties [Zelkha98] [Aarts99] [Streitz99].

Privat and Streitz define AmI as

> "a vision of the (not too far) future where *intelligent* or *smart* environments and systems react in an attentive, adaptive, and active (sometimes even proactive) way to the presence and activities of humans and objects in order to provide intelligent/smart services to the inhabitants of these environments." [Privat09]

Augusto defines AmI as

> "a digital environment that proactively, but sensibly, supports people in their daily lives." [Augusto10a]

Ambient Intelligence is enabled by the advances in multiple domains, such as networks, sensors, human-machine interfaces, artificial intelligence and robotics [Augusto10b]. These domains confluence to provide flexible and intelligent services to users acting in their environments. However, despite the provided intelligence, users may seek direct interaction with the system to indicate preferences and needs, so usability remains as an important AmI issue [Augusto10b]. Especially in buildings, research has shown that user interaction is necessary to achieve the desired building performance and provide for the comfort of occupants [Hartkopf86] [Choi10] [Daum11] [Gu11].

2.3. Smart Environments

A *Smart Environment*, also called *Smart Space*, combines ubiquitous computing with reasoning capabilities to create a human-centered system that is embedded in physical spaces and allows for adaptive behavior as the context changes [1]. Motivations for making the environment *smart* can be user comfort, energy savings, safety, security, protection of resources (e.g. hardware/material) and others. We model a *Smart Environment* as a subclass of an *Instrumented Environment*. As shown in the UML class diagram in figure 2.1, an instrumented environment consists of actuators and sensors. A thermometer is an example of a sensor. A *fixture*, such as a light fixture, is considered to be both, a sensor and an actuator, because it can be operated and provides a current status at the same time . The user is

[1] Definition adapted from [Cook04]

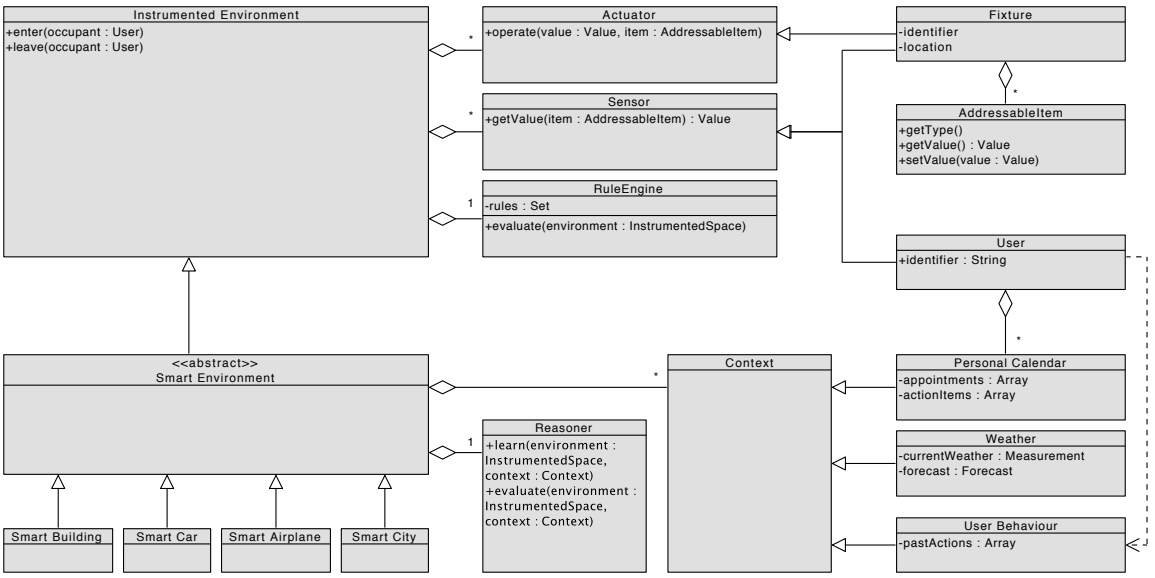


Figure 2.1.: Main model of smart environments as UML class diagram.

part of the environment and is modeled as a sensor. For example, the perceived temperature of a user might be very different to the measured temperature of a thermometer. This may lead the user to take action and control a thermostat. An instrumented environment also contains a rule engine which follows basic if-this-then-that rules. For example, a car can be modeled as instrumented environment if rain sensors are integrated to operate the windshield wiper automatically or light sensors turn on or off the lights of the car. Similar basic systems in buildings are common, e.g. to turn off the central heating unit during the night.

A *Smart Environment* is a subclass of an instrumented environment and adds additional aspects. It does not only provide some service automation, but furthermore learns and adapts its behavior during use. It is characterized by incorporating more complex reasoning capabilities to enable a flexible and adaptive behavior, taking into account the context of the environment as well as the context of the user. The transition between an instrumented and a smart environment may sometimes be smooth or parts of an environment may be smart while others are just instrumented. This can be seen in Aldrich's hierarchical classification of smart homes, which is discussed in section 2.3.1 below. Notably, a smart environment is modeled as an abstract class, meaning that it may only exist in the form of a more specific subclass. The UML class diagram in figure 2.2 shows examples of these subclasses. As these subclasses can be very different in their form, context and user be-

havior, we suggest to consider them more specifically, rather than trying to find solutions where one fits all. The focus of this dissertation is on smart buildings.

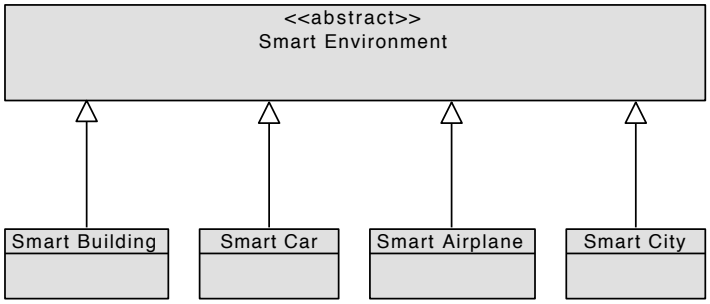


Figure 2.2.: Taxonomy of Smart Environment as UML class diagram.

There is also the notion of a *smart device*, *smart object* or *smart fixture*. We treat these terms as synonyms for embedded devices that are aware of their context to optimize their behavior. In turn, a smart environment contains these smart devices and supports the orchestration among them.

2.3.1. Smart Buildings

This dissertation focuses on smart buildings, a specific subclass of smart environments. A smart building serves as superclass for *smart home*, *smart office* and *smart factory*. This research aims at enabling multimodal controls in various kinds of smart buildings. The subclasses of smart building, as shown in figure 2.3, are explained below.

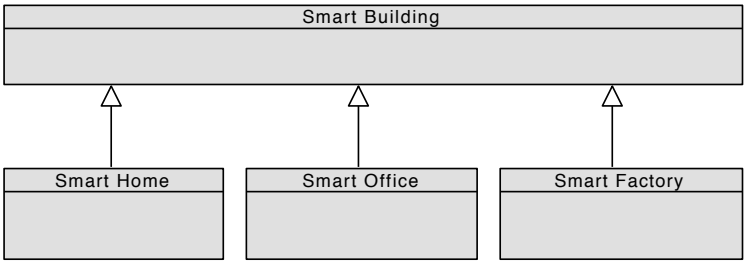


Figure 2.3.: Taxonomy of Smart Building as UML class diagram.

Smart Home. Aldrich has provided a well-known definition for smart homes as follows below.

> "A *Smart Home* can be defined as a residence equipped with computing and information technology which anticipates and responds to the needs of the occupants, working to promote their comfort, convenience, security and entertainment through the management of technology within the home and connections to the world beyond." [Aldrich03]

In addition to Aldrich's definition, convenient controls and energy savings are interwoven in any smart building because the energy saving mechanisms are working for the occupants comfort, too. Commercial products, such as NEST™ and TADO™ have been sold successfully with the promise to optimize the energy consumption in people's homes. However, despite the successfully deployed smart home technologies, Weiser's idea of ubiquitous computing lies beyond merely adding new stand-alone products to smart homes [Weingarten10]. This makes it important to give proper consideration to the implications in the homes of the future and in particular, to look at the usability questions related to these implications. Aldrich states that the insufficient attention being paid to usability of smart home technology is one of the obstacles in consumer take-up of smart homes. He thinks that it seems unlikely that attitudes are about to change in the supplier industry [Aldrich03]. The research described in this dissertation considers multimodal controls as essential part of the solution in future buildings.

Aldrich has looked at the emerging phenomenon of the smart home from a perspective of a social researcher. This underlines the difference between the various subclasses of smart buildings, as shown in figure 2.3. Aldrich stresses that the home is a *quintessential human place* [Aldrich03]. In particular, he emphasizes the historic role of women in homes and the opinion that they have long been disenfranchised from the development of the domestic technology, playing only little part in the design process and being viewed as passive consumers only [Aldrich03].

Aldrich has seen that the level of *smartness* brought into buildings and the functionality available to users can be very different between homes which simply contain smart appliances, and those which enable flexible and adaptive behavior as the context of the environment changes. This matches with our definition of *instrumented environments* compared to *smart environments* in section 2.3, as this transition can be smooth. Therefore, Aldrich came up with five hierarchical layers of smart homes [Aldrich03].

1. **Homes which contain intelligent objects.** Homes contain single, stand-alone appliances and objects which function in an intelligent manner.

2. **Homes which contain intelligent, communicating objects.** Homes contain appliances and objects which function intelligently in their own right and exchange information between one another to increase functionality.

3. **Connected homes.** Homes have internal and external networks, allowing interactive and remote control of systems, as well as access to services and information, both from within and beyond the home.

4. **Learning homes.** Patterns of activity in the homes are recorded and the accumulated data is used to anticipate users' needs and to control the technology accordingly.

5. **Attentive homes.** The activity and location of people and objects within the homes is constantly registered, and this information is used to control technology in anticipation of the occupants' needs.

Aldrich's classification highlights different levels of communication, distinguishes systems which are able to learn and differentiates homes which maintain constant awareness of occupants and objects [Aldrich03]. The classification is hierarchical because each level provides some increase in functionality from the user's perspective. From a technical perspective, each level generally depends on the infrastructure of the previous level. The issue of control over appliances and systems in the home emerges as an underlying theme. Moving up the hierarchical classification, the control systems range from the simplest switching mechanisms to highly complex systems. The opportunity for occupants to delegate control to the technology increases correspondingly. Aldrich suggests that the fifth level of smart home, the *Attentive Home*, with its potential for flexibility, proactivity and responsiveness to the user, enables the transition to homes that are substantially different to the current ones. [Aldrich03]

Smart Office. Compared to a smart home, smart offices aim at raising the occupants productiveness by providing the most comfortable work environment. Additionally, intelligent and personal control opportunities support energy savings in an environment where the typical occupant cares less about the energy consumption. This is due to the fact that home owners have to pay their energy consumption themselves while office workers are usually not accounted for their consumption in the office.

With the *Robert L. Preger Intelligent Workplace*, Hartkopf, Loftness et al. have shown an integrated approach to design and engineering of intelligent buildings [Hartkopf97]. The

Figure 2.4.: Intelligent Workplace interior.

living laboratory at Carnegie Mellon University in Pittsburgh, PA is also a *lived-in* laboratory, housing the people of the Center for Building Performance and Diagnostics. It has been used as a test bed for a number of experiments in this research. The Intelligent Workplace (seen in figure 2.4) incorporates numerous architectural features that provide for the health, productivity and comfort of its occupants, and the sustainability of its construction and operation. An overall objective is to significantly reduce the energy requirements for its operation in comparison to a conventional building space [Hartkopf97]. To achieve this goal, it includes ample natural lighting, natural ventilation with operable windows, and water-based cooling/heating supply.

Another example of a smart office has been shown by Streitz et al. in the i-LAND project [Streitz99]. It constitutes a vision of future workspaces with a focus on cooperative work of dynamic teams and changing requirements. The i-LAND project demands and provides new forms of human-computer interaction and computer-supported cooperative work. Streitz et al. augment room objects with information technology, e.g. resulting in an interactive electronic wall, an interactive table and two computer-enhanced chairs. [Streitz99]

Le Gal et al. [Gal01] present a smart office environment called *SmartOffice*. Through user monitoring, SmartOffice anticipates user intentions and augments the environment to communicate useful information. They envision computers helping users in their ev-

eryday tasks. The smart environment can interact with users through voice, gesture, or movement and can anticipate their activities. [Gal01]

The Stanford Interactive Workspaces Project has created and studied new technologies for integrated multi-person, multi-device collaborative work settings [Johanson10]. In addition to their primary *iRoom* testbed, they have deployed a number of interactive workspaces at Stanford university and other institutions, and evaluated their use in educational settings. The core technologies in these spaces provide a suite of tools for integration and interaction for co-located collaborative work. It features large shared and walk-up displays using touch/pen interaction. There are three major aspects to their research: Interaction design for shared computer-augmented spaces, robust flexible infrastructure for integration and empirical studies of collaborative work. [Johanson10]

Smart Factory. In the industrial sector, the *Smart Factory* has been coined as a term in Industry 4.0, which is a collective term that embraces manufacturing technologies for automation and data exchange. Within a smart factory, cyber-physical systems (explained in section 2.4) monitor physical processes, create a virtual image of the physical world and make decentralized decisions. Cyber-physical systems communicate via the internet of things (explained in section 2.5) and cooperate with each other and with humans in real time. In Industry 4.0, the participants of the value chain offer and consume both, internal and cross-organizational services. The overall goal is to use intelligent analytics and adaptive systems to improve the supply chain efficiency and reduce problems like defects, downtime and waste. Hermann et al. propose six design goals for smart factories [Hermann15].

1. Interoperability: The ability of cyber-physical systems and humans to connect and communicate with each other via the internet of things.
2. Virtualization: A virtual image of the smart factory which is created by linking sensor data with virtual plant models and simulation models.
3. Decentralization: The ability of cyber-physical systems within smart factories to make decisions on their own.
4. Real-Time Capability: The capability to collect and analyze data and provide the derived insights instantaneously.
5. Service Orientation: Offering and consumption of services.

6. Modularity: Flexible adaptation of smart factories to changing requirements by replacing or expanding individual modules.

These designs are implemented and tested in initiatives like *Smart Factory KL e.V.* in Germany with well-known partners such as Siemens, Bosch, Johnson Controls, Cisco, BASF, Continental and others [2].

2.3.2. Context Awareness

Context awareness is a key aspect of smart environments. The research domain is situated in the cross-section of ubiquitous computing, artificial intelligence and human-computer interaction [Augusto10b]. We use Abowd's definition of *Context* as the basis of our research.

> Context is "any information that can be used to characterize the situation of an entity (...) relevant to the interaction between a user and an application, including the user and the application themselves" [Abowd99].

That means, any information source can be considered as context as long as it provides relevant knowledge to handle communication between the user and the system. According to the definition, the user itself is also considered to be part of the context. With the advances in mobile technologies and smartphones, context information has become accessible when the devices are moving around with their users and store all kinds of personal context information, such as current location, weather, calendar, recent personal communication such as SMS, e-mail, social networks and others. To make context information and services accessible and manageable, Dey and Abowd present a context-based infrastructure for smart environments to support developers in building context-aware applications and services [Dey00]. Their framework abstracts the details of context-sensing and provides off-the-shelf components for context integration.

To act in a context-specific way, it is crucial to complete the transition from an *instrumented* to a truly *smart* environment. For example, in a building, an HVAC system may detect if a human is present but to react appropriately and to adjust the temperature to a comfortable level, it actually needs to consider the overall context. An occupant who is reading a book might require a warmer environment than someone being engaged in a

[2] Smart Factory KL e.V., http://www.smartfactory-kl.de

gymnastic exercise. This, however, requires an advanced form of spatio-temporal reasoning, since such classification is often dependent on when and where the behavior occurs [Augusto10b]. Therefore, the realization of location-based services has been a major interest of the research community.

Location-based Services

While GPS with its average accuracy of 15 meters[3] [McNamara08] enables location-based services in outdoor conditions, the tracking of occupants within buildings remains to be challenging. This is mostly due to the fact that much more precise location information is required indoors. In the HVAC example presented above, the system would not only need the information that an occupant is currently located in the living room but actually sitting on the couch. Technologies and techniques around Bluetooth LE, iBeacons, NFC, W-LAN triangulation, ultrasound positioning [Lazik12], QR codes and image recognition have been used to approach this issue.

Bauereiss and Peters have shown the integration of different types of location information sources using the blackboard architectural pattern [Bauereiß13]. However, a smart environment must not only be able to *determine*, but also to *predict* the location of an individual [Cook04]. This is of special importance, when mechanical processes are involved like in a smart building, where the rooms are warmed up appropriately as soon as an inhabitant returns. However, the heating process is not instantaneous and may involve a latency. Therefore, current approaches correlate multiple context sources and incorporate the history of past locations for the prediction [Cook04]. A second essential aspect of location context is the knowledge of the composition and interior of the building. Ontologies provide an effective representation for this kind of context and reasoning. For example, an ontology could know that a bluetooth beacon is left of the TV and that the user is left of the beacon and then provide the reasoning that the user is left of the TV. The more context information the ontology holds, the more sophistic reasoning it allows.

For the intuitive control of smart buildings, Brummit et al. have shown in their wizard-of-oz studies that a significant amount of users assumes the system to have spatial knowledge about the room and the user [Brumitt00]. From 198 performed speech commands, 16% used a reference to an object to refer to a light (*"the light above the couch"*), 30% used an area reference to refer to a light (*"all the lights in the living room"*), 23% included relative terms such as *"left"*, *"right"*, *"back"*, *"front"*, and 18% used an indirect reference (*"this*

[3] heavily depends on the overall conditions

light", *"all on this side"*, *"over there in the corner"*) [Brumitt00]. This shows the importance of context-aware and location-aware services in smart buildings.

2.3.3. Energy Awareness

Today's buildings still suffer from the problem that the energy consumption is intransparent for building occupants. Meters for electricity, water, gas and others usually reside in the basements of buildings, inaccessible for real-time energy feedback and without any detailed context information. Studies have shown that intransparency has led to a massive underestimation of people's energy consumption. When Poyser asked people how much energy a dishwasher would consume, it resulted on average in an underestimation by a factor of 800 times less than the actual consumption [Poyser13]. That entails the risk that energy is wasted because it is not even known that it is actually consumed. We assume that 30 - 40% of the energy in buildings could be saved without loss of comfort for the occupants.

One way of making energy visible in buildings is by using smart meters. A smart meter is an electronic device that records consumption of energy in specific intervals and communicates that information to the occupant and the utility for monitoring and billing purposes. Based on a European regulation (EDL 2006/32/EG), since 2010, the German Federal Network Agency (Bundesnetzagentur) has obligated providers to install smart meters in all new buildings. This is of particular importance since the consumption in identical homes, even those designed to be low-energy dwellings, can easily differ by a factor of two or more depending on the behavior of the inhabitants [Darby06]. Research has shown that providing feedback can reduce energy consumption by up to 20% and may even change habits, which results in long-term savings [Darby06] [Fischer08]. Considering that 19% of the total U.S. energy consumption in 2013 originated from the commercial sector [DOE13] and people are spending considerable time in their offices, raising awareness among them and giving appropriate feedback is vital to reducing the overall energy consumption. Beheim and Peters have shown an approach to raise energy awareness in smart buildings through personalized real time feedback on mobile devices [Beheim15]. The study showed that presenting the current consumption, split by appliances, increased their understanding of the consumption. For supporting energy savings, a fast and convenient access to control the fixtures turned out to be important.

Saving money can be a motivation for residents to reduce their energy footprint. However, this motivation in general does not exist in office environments, since employees

are usually not accounted for the energy bill of their office building. Pasat and Peters have shown an approach leveraging *gamification* to save energy collaboratively in offices [Pasat14]. Users can define sequences of energy-effective actions to be performed at once and send recommendations in the form of action sequences to their colleagues. By actively using the application, the users enter a competition while actively controlling their workplace in an energy-effective way and encouraging their colleagues for an environmentally aware behavior. The gamification strategy makes use of game elements such as points, badges, levels and high scores. The behavior of the users is rewarded with points and/or stars. By accumulating points, they can achieve high scores [Pasat14].

2.3.4. Disability Awareness

In their *World report on disability* [WHO11], the World Health Organization of the United Nations has shown that over a billion people, that is 15% of the world's population, have some form of disability. The rates of disability are increasing due to population ageing and increases in chronic health conditions, among other causes. Since the life-time expectancy has increased significantly, in particular in developed countries, a considerable increase in the percentage of elderly people among the population becomes visible [Bengtsson00]. Therefore, the requirements for assistance to elderly people are becoming more important and there is an increasing concern about the ability of elderly people to pursue an independent life in their preferred environment [D'Andrea09]. *Ambient Assisted Living* (AAL) aims at supporting this objective for both, elderly and disabled humans by applying technological innovations of ubiquitous computing. We consider the term *ambient* as a synonym for ubiquitous and unobtrusive.

It has been shown that people with motion impairments often prefer natural user interfaces, because these are "customizable, comfortable, and do not require user-borne accessories that could draw attention to their disability" [Augusto10b]. Natural user interfaces are discussed in section 2.6. Having interfaces that are intuitively usable, i.e. being easily learnable, becomes important when taking into account the decreasing mental capacities while humans grow older. Intuitive user interfaces are discussed in section 2.7. The variety of possible physical disfunctions makes it even more necessary, to have a customizable user interface in smart buildings because each individual user might need an interaction model which suits to the individual physical capabilities. MIBO approaches this need by making interaction models in buildings configurable for individual users and user groups.

Another aspect of the ageing society is the issue of providing sufficient medical treat-

ments with a decreasing density of medical supply. *E-Health* aims at providing doctors with information about the physical conditions of a remote patient. Eventually, smart homes allow for physical exercises, being supervised by a doctor from remote. Friedrich, Hiesel, Peters et al. have used serious games for home-based rehabilitation after strokes [Friedrich15]. The objective of the game approach is to enrich the training experience and establish a higher level of compliance to prescribed exercises, while maintaining a supportive training environment as found in common therapy sessions. The system provides a collection of mini games based on rehabilitation exercises used in conventional physical therapy, monitors the patient's performance while exercising and provides clinicians with an interface to personalize the training. The clinician can set the current state of rehabilitation and change the playable games over time to drive diversification. An early stage case study offered positive indications towards this concept.

2.4. Cyber-Physical Systems

The term *embedded system* is used to describe engineered systems that use sensors and actors to interact with their environment for a dedicated purpose [Lee08]. The envisioned transformation that makes an embedded system to become a *Cyber-Physical System (CPS)*, can be achieved by networking embedded systems [Lee08] [Rajkumar10] to overcome the gap between physical objects in the real world and their representation in information technology. CPS allow changing the control logic of hardware devices at runtime and outsource computational-intense tasks to remote systems. This shift of control logic provides CPS with a considerable increase in computational capabilities and enables the integration of the overall context of the user and the environment.

We assume that Cyber-Physical Systems will have a significant impact on future buildings, as they will be embedded in all types of objects and structures. An example of the transition from embedded systems to CPS can be found in thermal control systems. A traditional embedded HVAC system follows well-defined paradigms of a feedback loop, comparing the actual temperature with the desired setpoint and regulating the flow of a heat transfer fluid, to maintain the correct temperature. On the other hand, Cyber-Physical Systems are networked to include weather forecasts, personal calendars, remote control opportunities for users and apply machine learning techniques to learn about the user behavior. By taking better decisions, networked building control systems could significantly improve the energy efficiency of buildings in the future. Furthermore, with many build-

ings becoming producers of energy [Ayoub13], Cyber-Physical Systems of multiple buildings may span a *smart grid* to coordinate the production of power with large numbers of small power producers and consumers. Eventually, the transformation from *instrumented* buildings to *smart* buildings is accompanied by the transition from embedded systems to smarter, connected devices called Cyber-Physical Systems.

2.5. Internet of Things

The term *Internet of Things* (IoT) was originally coined for logistic applications by Kevin Ashton in 1999 to describe the link of radio-frequency identification (RFID) with the internet [Ashton09]. Afterwards, the term has been adopted and generalized for other application domains. It has been added to the Oxford dictionary by Stevenson (Ed.) as a new word in 2013:

> The *Internet of Things* is "a proposed development of the internet in which everyday objects have network connectivity, allowing them to send and receive data." [Stevenson13]

Although the initial application of IoT was focused on logistics, it has an impact on building technology as well. The IoT enables the notion that every fixture in a smart building becomes an addressable object, accessible through the internet. This is a fundamental requirement for realizing *Ubiquitous Computing* and thus opens up new possibilities for indoor environmental quality and energy savings in buildings. Hersent states in his book *The Internet of Things: Key Applications and Protocols* [Hersent12] that a reason for the emerging IoT in buildings originated in the multitude of proprietary, incompatible protocols and standards (see section 4.1.3). He assumes that the need for a common networking technology running over any physical layer, like the Internet Protocol (IP), has become clear [Hersent12].

Internet blogs and marketing activities use the term *Internet of Everything* (IoE) or *Internet of Anything* (IoA) to underline the trend of adding connectivity not only to *physical things* but to just about *everything* (e. g. people, processes, data) [4]. However, we consider all these terms to be equivalent and stick to the most common term *Internet of Things* (IoT).

[4] Source: Cisco, http://www.cisco.com/c/dam/en_us/about/ac79/docs/IoE/IoE-AAG.pdf

2.6. Natural User Interfaces

Graphical user interfaces (GUIs) follow the *WIMP* paradigm, using windows (W), icons (I), menus (M) and a pointing device (P). These interfaces are widespread and well-known in the most popular operating systems, such as Microsoft Windows and Apple OS X. However, these paradigms have only few in common with human-human interaction, which is mostly based on speech, gestures and eye-contact. *Natural User Interfaces* (NUIs) aim at overcoming the traditional GUIs with more natural ways of interaction by adopting modalities from human-human interaction to human-computer interaction (HCI). Since NUIs are effectively invisible, they can be used to realize ubiquitous and invisible computing, as envisioned by Mark Weiser. Some researchers and industry leaders assume that NUIs will be the successor of GUIs in the same way as GUIs have replaced CLIs (command line interfaces) [Lu12].

Natural User Interfaces include gesture-based interfaces, voice-based interfaces and gaze-tracking. If only one modality is involved at a time, we refer to *unimodal* user interfaces. If multiple modalities are involved in an interface, we refer to *multimodal* user interfaces.

2.6.1. Unimodal User Interfaces

A unimodal user interface involves one modalitiy at a time. In the following subsections, we describe user interfaces that are based on the modalities gestures, speech and gaze.

Gesture-based User Interfaces

Gestures are an expressive form of interaction between humans in our everyday lives, from waving, pointing and clapping up to filigree finger gestures, that can often make a delicate difference in their meaning. Gesture-based interfaces can be partitioned into touch-screen gesture interfaces (well-known in recent smartphones) and body gesture interfaces, involving the hands and arms of the human. We consider body gestures to be most similar to the natural interaction among humans. Thus, in this dissertation, we always refer to body gestures when we discuss gesture-based user interfaces. These interfaces allow for a thoroughly invisible computing because they do not require a touch screen or any visible computing device. The recognition of gestures can be performed using Hidden Markov Models (HMMs), as applied by Kela [Kela05]. HMMs are a statistical modeling technique for the analysis of a time series with properties that change over time and are widely used in speech and hand-written character recognition as well as in gesture recognition [Kela05].

A second approach for gesture recognition is the Dynamic Time Warp (DTW) algorithm, as shown by Liu [Liu09] and Lou [Lou13]. DTW is a non-statistical recognition algorithm and can be used to avoid the long-time training process required by HMM.

When designing gesture-based interfaces, the selection of effective and natural gestures has to be examined for the multitude of tasks they could be used for. Kela et al. have performed two user studies related to the selection of gestures [Kela05]. The first study compares the gesture modality to other modalities such as speech, RFID-based tangible objects, laser-tracked pen and PDA stylus. The study revealed that gesture commands were found to be natural especially for commands with spatial association. In addition, the authors point out the effectiveness of combining modalities with each other.

The second user study by Kela et al. investigated suitable gestures for controlling a studio equipped with a TV, VCR and light fixtures [Kela05]. Kela stresses that the topic is very wide in scope since there are a large number of suitable gestures for certain tasks, as well as many tasks that could potentially be performed using gestures [Kela05]. The study showed that the same gestures were often used for similar commands, for instance an up-movement for lights brighten, TV volume up, VCR play or a down-movement for lights dim, TV volume down, VCR stop. The authors conclude that the users would like to control different devices using the same basic gestures [Kela05]. Up/down movements with the hands was a gesture type that was performed continuously by users. *Up* was used for commands such as *on*, *increase*, *raise*, *brighten*, *start*, *play*. In contrast, *down* was commonly used for *off*, *decrease*, *lower*, *dim* and *stop* commands. Kela's study also revealed that different people prefer different gestures for the same tasks. This underlines the requirement to personalize interaction models. [Kela05]

Assuming the same gesture (e.g. up/down) is used as a kind of polymorphism for different fixtures, there is a need for a method to select the addressed fixture associated with the command. Dan Saffer suggests the pointing gesture in his book *Designing Gestural Interfaces* to select/activate objects [Saffer08]. He emphasizes that pointing is the most natural gesture for selection.

In 2009, Bhuiyan and Picking published a review of gesture-based user interfaces to identify trends in technology, application and usability [Bhuiyan09]. They summarized the history starting from early handwriting gestures on the stylus device in 1986 up to current commercial products such as the Nintendo Wii and recent research up to 2009. The tabular summary is enclosed in Appendix B on page 133ff. Bhuiyan and Picking conclude

that gesture-based user interfaces afford realistic opportunities for specific application domains. They consider natural user interfaces using gestures as appropriate for current and future ubiquitous and ambient devices. [Bhuiyan09]

Our hypothesis is that gesture-based interfaces provide an effective approach to control fixtures in buildings. We have shown an intuitive controller for home and office environments using gestures, performed on a smartphone [Peters11]. Therein, the user simply points the smartphone at target objects and makes specific gestures. The implementation of the point-and-gesture control uses the magnetometer, gyroscope and accelerometer built into the most recent smartphones [Peters11]. Kela et al. proposed an accelerometer-based gesture control for a studio room, which is able to recognize gestures (e.g. drawing a letter in the air) and link them to a specific command [Kela05]. Yang et al have shown a human-friendly HCI method for the control of home appliances using two installed cameras in a room. Their approach is specifically tailored to old and disabled people [Yang07]. Tsukada and Yasumura presented the UBI-FINGER, a device for finger input gestures in ubiquitous environments [Tsukada04]. Wilson and Shafer have described *XWand*, a wand-similar device to control appliances using simple gestures [Wilson03].

Voice-based User Interfaces

Natural language is an expressive method for communication and interaction between humans. While graphical user interfaces include a visual display, keyboard and mouse, voice-based interfaces use spoken natural language for human-computer interaction. The technology to allow humans to communicate with computers by speech has become accessible in many computing devices. Commercial implementations, such as Apple Siri, Google Now and Microsoft Cortana are widely used on today's smartphones.

Speech recognition enables the recognition and translation of voice signals into text by computers. It can be modeled as a pipeline of processes, including pretreatment, feature extraction, pattern matching and training [Lu12]. Pretreatment includes *pre-emphasis* to improve the higher frequencies in the voice signal input, *framing and windowing* to divide the signals into time fragments and *voice activity detection* to determine the start and endpoints of a signal [Lu12]. Multiple methods for feature extraction can be applied, such as Mel Frequency Cepstrum Coeficients (MFCC), to extract the characteristics of the voice signal reflected by the energy distribution in the frequency domain [Lu12]. Hidden Markov Models (HMMs) have also been applied to feature extraction in speech recognition [Lu12]

[Kela05]. Erman et al. modeled their Hearsay-II speech recognition system using independent processes to achieve cooperative problem-solving behavior [Erman80]. Their blackboard architecture reconstructs an intention from hypothetical interpretations at various levels of abstraction. Hearsay-II successfully integrates and coordinates all of the independent activities to resolve uncertainty and manage complexity [Erman80].

Peters [Peters11], Brummit [Brumitt00] and many others have tested the usage of voice-based interfaces in smart environments and showed the effectiveness of voice interfaces to control fixtures. In a wizard-of-oz experiment, Brummit found out, that the test persons used the simple words *on*, *off*, *bright* and *dim* in 160 of 198 commands [Brumitt00]. With the release of HomeKit, Apple has recently enabled voice-based interactions in instrumented buildings for a larger audience using similar command words as identified by Brummit.

Eye-based User Interfaces

Eye-based user interfaces enable another way to interact with computers by tracking the gaze of the eyes. With the introduction of smart glasses, the breakthrough of eye-based user interfaces has eventually become feasible. We assume that these user interfaces will play a more popular role in human-computer interaction in the near future. When Brummit asked people to control fixtures in a room in his wizard-of-oz study, he noticed that "there was one action that everyone seemed to do when referring to lights: they looked at the light they wanted to control" [Brumitt00]. Brummit states that in only 9% of the tasks, people never looked at the fixture for which they issued commands. He concludes that the gaze of the eyes is a useful modality to aid in disambiguation with speech interfaces [Brumitt00].

In the COGAIN project [Corno10], eye-based home automation has been tackled by exploiting state-of-the-art technologies. Integration between the eye-based interfaces and the wide variety of fixtures that may be present in a smart building is achieved by the implementation of a Domotic House Gateway that adopts technologies derived from the semantic web to abstract devices [Corno10]. The original focus of the project was to develop solutions for users with severe motor disorders, such as Myotrophic Lateral Sclerosis, Motor Neurone Disease, Multiple Sclerosis, Cerebral Palsy, Spinal Cord Injury, Spinal Muscular Atrophy, Rett Syndrome, Stroke and Traumatic Brain Injury [Corno10]. Eventually, the only way of interaction for these users may be the use of their eyes. The potential numbers of affected users in the European Union alone has been estimated as approximately three

million [Corno10]. We assume that eye-based interfaces can provide an effective interaction modality not only for users with movement disorders. This development has to be accompanied by solutions to securely maintain the user's privacy.

2.6.2. Multimodal User Interfaces

Multimodal user interfaces (MMUIs) aim at using naturally occurring forms of human behavior which incorporate two or more combined user input modalities at the same time [Sun06] [Dumas09]. The first MMUI *"Put-That-There"* has been presented by Bolt when he combined speech and gestures to allow users commanding simple shapes over a large screen surface [Bolt80]. Several recognition-based technologies process and integrate the involved modalities in a coordinated manner. MMUIs can be used to enable multimodal input (human-to-computer) as well as to provide multimodal output (computer-to-human). The output may be given by audio/visual cues, speech or haptic feedback. Furthermore, the output may be given through multiple devices, such as a smartphone, watch, via speakers in a room, mounted displays or projections. Figure 2.5 shows a model of the multimodal human-computer interaction.

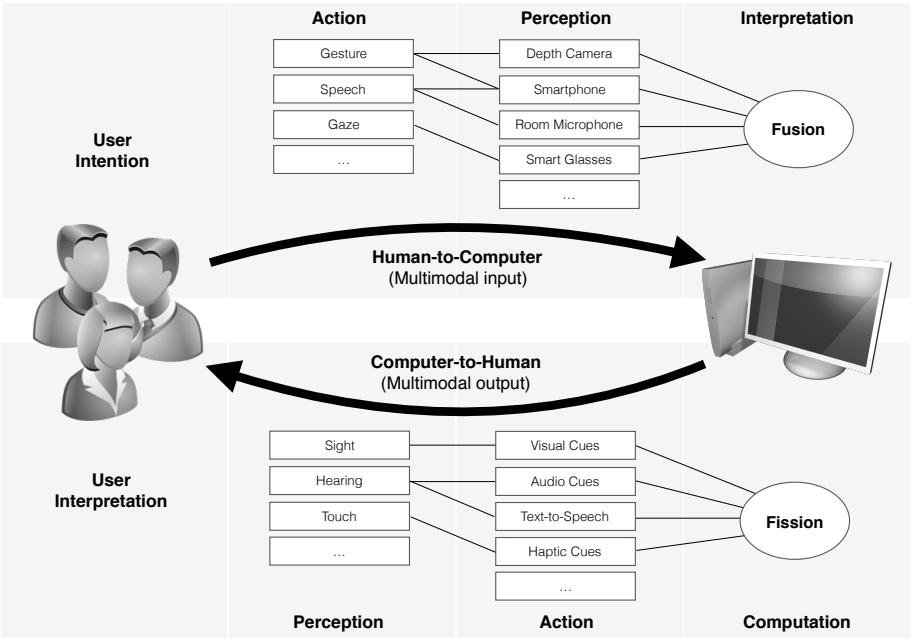


Figure 2.5.: Multimodal human-computer interaction model. Adapted from [Dumas09].

Multimodal user interfaces were initially considered to be more efficient than unimodal interfaces [Dumas09]. However, Oviatt's evaluations showed that multimodal interfaces speed up task completion only by 10% [Oviatt97] [Oviatt02] [Oviatt03]. Hence, Dumas argues that efficiency should not be considered to be the main advantage of multimodal interfaces [Dumas09]. Though, multimodal interfaces have been demonstrated to improve error handling and reliability. The users in Oviatt's studies made 36% fewer errors with a multimodal interface than with a unimodal interface [Oviatt97]. Finally, MMUIs provide improved support for users' preferred interaction style, since 95% of users stated to prefer multimodal interaction over unimodal interaction [Oviatt97]. D'Andrea adds that multimodality also improves accessibility by encompassing a broader spectrum of users, enabling those of different ages and skill levels as well as users with disabilities to access computers [D'Andrea09].

Examples from implementations of innovative control systems in buildings have also shown that a unimodal interface alone may not be sufficient [Peters11] [Wilson03]. Although speech, for example, would clearly have enough expressive power, relying on speech alone can be difficult in practice regarding efficiency [Wilson03]. The wizard-of-oz study, described in section 3.2, supports these findings. While speech may have advantages, e.g. to set a temperature value in a room, spatial referencing to a very specific fixture (e.g. "the second left light fixture, seen from the north-west window") feels cumbersome for users.

Multimodal Integration

To allow for the combination of multiple modalities, the various input streams have to be integrated to meaningful information. Multimodal integration is a key technical challenge for multimodal interaction systems, since the meanings of input streams can vary according to context, task, user and time [Turk14]. Modalities, such as voice, gestures, eye gaze, facial expression and haptics, have very different characteristics and may not have straightforward ways to integrate. The temporal dimension adds to the complexity, as different modalities may have different temporal constraints and different signal and semantic endurance [Turk14].

Multimodal integration consists of two aspects, *multimodal fusion* to integrate multimodal input streams and *multimodal fission* (also called response planning) to distribute output streams over multiple modalities. This dissertation is mainly focussed on multi-

modal input, rather then output. The presented MIBO framework uses a reference architecture for multimodal fusion of modalities using the blackboard pattern. The multimodal fission has been shown by Perroud et al. in an approach for the generation of context-aware multimodal feedback in smart environments [Perroud11a].

A key issue in multimodal fusion is the question, how and when modalities should be integrated [Turk14]. Sharma et al. have proposed three distinguishing levels of abstraction for multimodal fusion [Sharma98].

- **Data-level fusion** implies the fusion of identical or tightly linked types of multimodal data, such as the fusion of two video streams being recorded from different angles. Data-level fusion works on the raw data, which means it has access to the most original form of it but is also highly sensitive to noise or failures [Hoste11]. Therefore, it usually entails some initial signal processing, such as noise filtering. Data-level fusion rarely deals with the semantics of data but may try to enrich or correlate data that is potentially going to be processed by higher-level fusion processes. [Hoste11]

- **Feature-level fusion** is applied on features extracted from the data rather than on the raw data itself. It typically applies to closely coupled modalities with different representations, such as speech and lip movement integration [Hoste11]. The data would be provided by a microphone, which is recording the speech and a camera filming the lip movements. In this case, the goal of the feature-level fusion could be to improve speech recognition by the combination of information from two modalities. Hoste et al. argue that feature-level fusion is less sensitive to noise than data-level fusion and conveys a moderate level of information detail [Hoste11]. Typical feature-level fusion algorithms include Hidden Markov Models (HMMs), Neural Networks and Dynamic Time Warping (DTW).

- **Decision-level fusion** derives interpretations based on semantic information and correlates data from loosely coupled modalities, such as speech and gestures. Additionally, semantic information from the feature level allows mutual disambiguation. However, since decision-level fusion relies on the quality of previously acquired semantic information, the information being available for decision-level fusion algorithms may be incomplete or distorted. Typical methods of decision-level fusion are meaning frames, unification-based or symbolic-statistical fusion algorithms. [Hoste11]

Multi-device user interfaces are a more specific subclass of multimodal user interfaces (MMUIs). It means that more than one recognition device is involved during the interaction. When we refer to MMUIs, we also include multi-device user interfaces. This is useful in smart buildings, as different modalities may leverage the strengths of different sensor devices. For example, gestures may be recognized by installed depth cameras, while speech may be captured by a wearable computing device, such as a smartwatch.

Despite the long list of related work and the advances in the research domain, such as fusion of heterogeneous data types, dialog management, map-based multimodal interaction, it is commonly believed that the field still needs further research to build reliable and useful applications [Dumas09] [Turk14].

2.7. Intuitive User Interfaces

Human-Computer Interaction (HCI) is a domain of computer science and seeks to enhance the interaction of humans with computers by improving the usability of computer interfaces. The term *usability* has been standardized in the 1980's as the *Ergonomic Requirements for Office Work with Visual Display Terminals* and has been enhanced continuously since then in the ISO 9241 standard. Therein, usability is defined as follows.

> "**Usability**: The effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments.
>
> **Effectiveness**: The accuracy and completeness with which specified users can achieve specified goals in particular environments.
>
> **Efficiency**: The resources expended in relation to the accuracy and completeness of goals achieved.
>
> **Satisfaction**: The comfort and acceptability of the work system to its users and other people affected by its use." [ISO 9241]

Additionally, studies reveal a new type of users who do not read manuals anymore but rather use their intuition to interact with systems [Novick06]. With Weiser's vision of ubiquitous computing [Weiser91], wherein a smart environment is interwoven with invisible computing devices, usability becomes even more important. Therefore, it is expected from developers to design interfaces that reduce the required cognitive load and are usable intuitively. In this dissertation, we use our own definition of "intuitive use" which is

loosely based on Naumann et al. [Naumann07] but highlights the relation to learnability and provides measurability for a defined level of effectiveness.

> A technical system is, in the context of a task, *intuitively usable* for a user, if the user is able to learn rapidly how to interact effectively with the system. The faster this learning process can be completed, the more intuitively usable is the system.

Norman claims that users are applying their conceptual model when interacting with devices [Norman90]. The conceptual model formed by a user depends on visible system components, the mapping between the visible objects of the system and the actions supported by these objects. Intuitively usable systems should provide a *natural mapping*, which enables a user to easily create a conceptual model.

> "A natural mapping is a mapping between controls and their result in the real world such that the mapping does not tax the user's memory when performing a task that involves the manipulation of these controls." [Norman90]

Our hypothesis is that intuitive user interfaces can be achieved by bridging the gap between the user's conceptual model and the system model. Intuitively usable tasks in a system provide natural mappings and thus, require minimal user documentation or no documentation at all. Intuitively usable systems help the user to build a correct conceptual model of the system by providing a user interface, which reduces the load on the short term memory (ideally not more than seven plus/minus two visible parts [Miller56]). They must also provide good feedback when the user performs an action and must accommodate errors by providing a robust user interface with a natural mapping even in error cases.

Our definition of intuitive use also relates to *effectiveness*, as it is defined in the ISO 9241 standard. As described earlier on page 29, usability goes beyond effectiveness by also incorporating the notions of efficiency and satisfaction. These aspects are important but we do not consider them as part of the definition of intuitive use. For example, a system might be familiar and learnable very quickly but still not efficient and satisfiable. A user interface might also incorporate *natural* modalities, such as speech and gestures, but still use them in a way so that they are neither intuitive, nor usable effectively or efficiently. Established graphical user interfaces (GUIs) may follow the "design rule of [...] visibility" [Norman90] which allows them to "be learned through exploration" [Norman90]. Natural

user interfaces lack this visibility and are therefore different to learn, e.g. by imitation. It is noteworthy, however, that it is not a binary decision to determine whether a system fulfills the metrics for intuitive, natural and usable.

This dissertation focuses on intuitive user interfaces for controlling smart buildings, short "intuitive controls in smart buildings". When we refer to "intuitive controls", we aim at all three categories of user interfaces – intuitive, usable and natural. To achieve these goals, requirements are described in the following chapter.

3. Requirements for Intuitive Controls in Smart Buildings

Based on our definition of intuitive, usable and natural user interfaces, we define requirements for multimodal controls in smart buildings. The elicitation of requirements follows a formative approach and is based on real user experiments. We present a methodology, using guessability study, wizard-of-oz experiment and expert reviews to explore the requirements by combining a bottom-up and a top-down approach. The results of our study support common assumptions about the use of multimodal interaction within smart environments. Next, we present visionary scenarios to define the scope and the involved support of MIBO for the particular target audience. Finally, functional and non-functional requirements are derived and compared to both, literature and our own studies.

3.1. Methodology

According to Cafaro et al., there are two different approaches for the creation of gesture interfaces -— bottom-up and top-down [Cafaro14]. We consider this distinction also for natural user interfaces. Bottom-up approaches are based on the idea of participatory design, where eventual users of a system assist in the design and development [Schuler93]. In top-down approaches, the designers decide on the suitable interactions of the user with the system. This can have several advantages for the user since subject matter experts can pay special attention to additional factors that are not evaluated in user studies. For example, the technical recognizability of interactions to increase the fault tolerance, or the exhaustion effects for humans, if interactions like gestures are repeated more often than tested with users.

We propose a hybrid approach, depicted in figure 3.1, to create intuitive user interfaces in smart environments. Thereby, users should be involved from the very beginning of the project, followed by expert reviews being given the final decision power to consider technical recognizability, human exhaustion effects and other external factors that influence

the usability of interactions. For bottom-up approaches with user involvement, we consider the combination of two experimental key methodologies – *Wizard-of-Oz Experiments* and *Guessability Studies*.

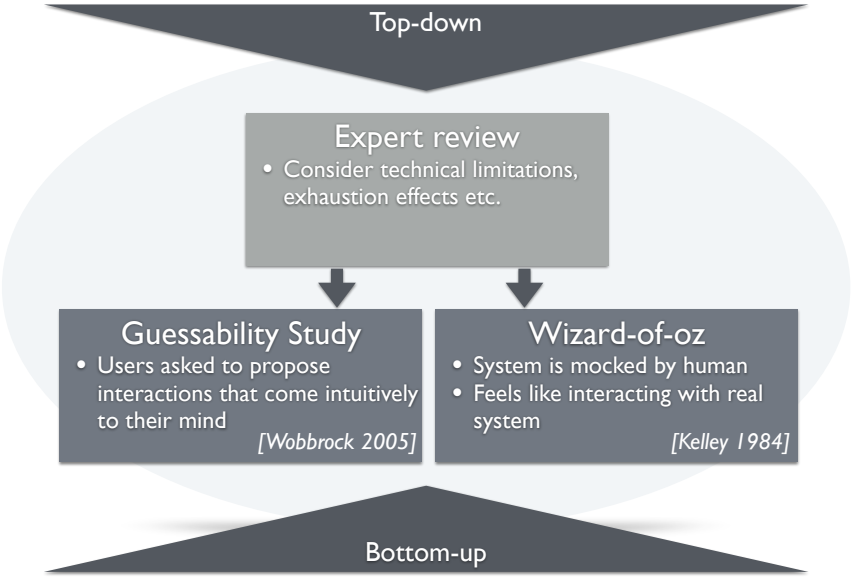


Figure 3.1.: Combination of bottom-up and top-down approaches.

Wizard-of-Oz Experiments. The Wizard-of-Oz technique has first been described by Kelley [Kelley84]. It is used in a very early stage of system development where users interact with a system that they believe to be implemented but that is actually being faked by a human. Since the implementation of multimodal systems is usually very costly [Augusto10b], it is desirable to assess the usability at design time, prior to implementation. This allows developers to get an early feedback on whether the proposed interface, possibly implying modalities such as gestures and speech, is effectively usable and understood by users. The actual implementation of the system is mocked by a human, the *wizard*.

Guessability Studies. In a guessability study, users are asked to propose an input symbol for a given output. For our use case, we ask a group of users to propose interactions for a certain task in a smart building. The purpose of these studies is to maximize the guessability of an interaction and help usability engineers to find intuitively usable interactions.

The method has been introduced by Wobbrock et al. [Wobbrock05] to maximize the guessability of symbolic input. For instance, a symbolic input might be an icon, a gesture or a speech command. Guessability of symbolic input is defined as the "quality of symbols which allows a user to access intended referents via those symbols despite a lack of knowledge of those symbols" [Wobbrock05]. The collected responses of the study-participants are then checked for conflict groups and scores are determined for the whole set of symbols.

We argue that the combination of these approaches increases the guessability of interactions and helps usability engineers find intuitive controls in smart environments. While guessability gathers the users intuitive ideas without forcing them to think in pre-defined system models, the wizard-of-oz technique mimes a realistic interaction flow for the user. Users are observed, while they are interacting intuitively without further instruction, and the system behavior is mocked by an invisible wizard. Furthermore, we emphasize that intuitively usable interfaces should always follow the notion of human-centricity, which refers to the following paradigm in developing and utilizing technology. Therein, the user is put in the center of the overall development to be served in a way that best offers the intended user experience of the application [Augusto10b].

3.2. Requirements Identification

Based on this methodology, we conducted a user study and asked test persons how they would control a smart room intuitively. The system was mocked by another human and responded appropriately to any given commands of the participants. Afterwards, the participants were questioned in a semi-structured interview. The study showed the advantages of deictic gestures compared to speech for commands with a strong spatial reference, such as switching on a specific light fixture, and for very simple commands when speech could be avoided. On the other hand, speech showed advantages for those commands without spatial reference, such as setting a temperature. Multimodal commands, incorporating both, speech and gestures, were most liked for more complex tasks.

Oviatt conducted several user studies [Oviatt97] on multimodal user interfaces. These tests were mainly focused on map-based systems, combining speech and pen input. She showed that users made 36% fewer errors with a multimodal interface than with a unimodal interface and that almost all users stated their preference of multimodal interaction

over unimodal interaction [Oviatt97]. However, the involved tasks and interactions in map-based systems using speech and pen are quite different to those for controlling fixtures in a smart building. Wilson and Shafer [Wilson03] showed a wand-like sensor device called *XWand*, to control appliances using simple gestures and speech in a smart environment. They show valuable multimodal interactions in smart buildings but did not further analyze preferred modalities for different sets of tasks. Carrino et al. have shown a multimodal interaction approach [Carrino11] for smart environments using a camera worn on the user's arm. Compared to our approach, their study does not investigate alternative interactions but rather focuses on the accuracy of detected gestures to test the performance of their system. Brumitt et al. have examined interfaces to control lights in homes of the future [Brumitt00] by conducting a similar combination of wizard-of-oz study with guessability approach. Speech and the multimodal combination of speech and gestures were among the preferred interaction modalities in their residential setting. In our research, we want to extend the study beyond the usage of lights to explore the requirements of different modalities for different tasks. Additionally, our study is not targeting a residential setting but rather an office environment. Anastasiou et al. present another user study [Anastasiou13], investigating gestures and speech in an ambient assisted living (AAL) environment, comparing preferences of German and Chinese participants. However, their focus is limited to the unimodal analysis of gestures and focuses on the support of elderly and disabled people.

3.2.1. Study Design

As described in section 2.7, we combined a guessability study with a wizard-of-oz experiment, in order to see users experimenting in a smart room and propose interactions on their own. We observed the users while they were interacting intuitively without any further instruction and mocked the system behavior using a wizard-of-oz (see figure 3.2). We suggest using this method for systems that focus on maximizing for the intuitive use. This actual experiment was conducted with ten participants in a room of an open office space, which was familiar to the test persons. The participants knew from their daily experience that there were no traditional buttons installed in the space. Usually, they use an interface on their desktop computer or a gesture-based smartphone controller (described in section 6.1) to control their lights and shading devices.

We tested nine different tasks, each in three modalities, with ten different users, which results in a total of 270 performed commands. The test users were equally distributed

Figure 3.2.: Test person interacting with the smart room.

regarding genders and originated from nine different nationalities. Each test person did the experiment alone, without seeing other persons taking part in the experiment. We introduced each participant to the room just by saying that this room was actually smart and that we had installed all kinds of sensors to sense anything they do to interact with the environment. Next, the persons had to complete the following tasks:

T1 Turn on a specific light fixture (among several).

T2 Turn off the same specific light fixture.

T3 Dim all four overhead lights to 70%.

T4 Turn on a fan.

T5 Turn off the same fan.

T6 Start the charging process of an iPhone.

T7 Turn on a task light at the desk.

T8 Turn off the same task light.

T9 Set temperature to approx. 19° C / 66° F.

We focused on ubiquitous tasks that are commonly performed in today's buildings in the areas of lighting, ventilation and thermal control. For generalizability, we aimed to cover different aspects of tasks, some with strong spatial reference (T1, T2) and some with less (T9, no visible thermostat in the room). Some also required additional parameters, such as a specific light level (T3) or temperature setting (T9). We also added a more complex command, requiring the user to start charging a smartphone (T6). First, we did not ask people to use any specific modality for the tasks but just observed what they would use intuitively in a smart room. Next, we asked them to use particularly speech, then gestures and last, the multimodal combination of speech and gestures. We never told them any particular speech or gesture commands to use but rather observed what they came up with (guessability study). Whenever there was a unique, understandable command given, our wizard controlled the environment accordingly (wizard-of-oz experiment). At the end of the experiment, we conducted a semi-structured interview, in order to evaluate and rank the different controls that the persons have used for their tasks. We also asked the people for the reasons of their answers.

3.2.2. Results

When our test persons interacted with the smart room to complete their tasks, they started with different modalities intuitively and even when we asked them to use specific modalities, they made use of them in quite different ways. Even though this fact is a finding on its own, in the following, we focus on the strongest observations that we could see on a considerable majority or even all test persons. It became visible that all of the test persons do not like to use speech alone for tasks with strong spatial reference, i.e. that refer to a specific fixture among several fixtures of the same type. Even though speech has sufficient expressive power, the usage of spatial commands (e.g. "the left fixture which is not next to the window") became awkward and often ambiguous. Additionally, when we asked about situations when participants would never like to use speech, 100% stated that they would not like to use speech in public situations, among people or in quiet office environments, such as the test environment. Three exemplary answers are shown below to explain the reasons:

- "People might think I am crazy."
- "These systems usually can't handle my strong accent, so it becomes frustrating."
- "Talking to fixtures felt weird."

While speech alone worked better for unique devices (T4/T5, "turn on/off the fan"), it became difficult whenever there were multiple of these devices (T1/T2). People preferred to use gestures then and pointed at devices. 100% of the persons favored the usage of gestures instead of voice for simple tasks, such as turning on a single light fixture. However, for more complex tasks like charging the iPhone, 70% of the participants ranked the usage of speech higher than gestures and 10% ranked them equal.

We then asked about an overall ranking of modalities for building controls. 70% of the persons gave higher points to unimodal gestures, compared to unimodal speech. This leads to the assumption that voice should be avoided for tasks where unimodal gestures are sufficient. This was backed up by the participants during the interview. 60% of the people gave their highest points to multimodal controls in their overall ranking. All of the other 40% stated that the multimodal combination was sometimes a bit too much but actually helps for more complex commands. This result reveals the necessity of thorough decisions when to use unimodal versus multimodal interaction. However, when people were not able to think of a gesture for a task, such as for setting the temperature, they could also not think of a multimodal command when speech was added. All persons used speech alone for this task, which seems appropriate. It was surprising, that despite of the well-known conversational style of speech assistants in recent smartphone operating systems, all our test persons used mainly just isolated words ("lights off", "turn on", "turn off"). 80% of the persons used relative commands for their thermal controls, such as "warmer" or "colder", even though we asked them to set the temperature to 19° C / 66° F.

We also noticed that all our test persons actually looked at the fixtures they wished to control. One participant even mentioned the wish by himself that the system should be able to track his gaze. With the emerging market in wearable computing and smart glasses, we assume that gaze tracking will be a powerful modality in the future for intuitive building controls.

To simplify the controls, users also expected a certain degree of context-awareness, such as remembering a control history. For example, people wanted to be able to refer to the same kinds of fixtures that they controlled recently (e.g. "turn on fan", then "turn it off again").

Next, we evaluated the differences between those persons who have used our phone-based gesture controls [Peters11] regularly within the last four years and those who were not used to that form of control. We observed that all of the persons who were used to gesture with their smartphone intuitively tried to use gestures with their hands as well

to control the environment, while the ones who were not used to these kinds of controls tried to look for traditional switches first. When they could not find any, they started to use voice instead. We could see that the performed in-air gestures looked similar to the ones being used in the phone-based gesture control. Interestingly, 50% of the persons explicitly preferred to hold a device, such as a smartphone, in their hand while performing these tasks, the others stated that they preferred to have no device because of simplicity. This raises the requirements to support different styles of supported user interactions (e.g. device-based and camera-based).

3.3. Visionary Scenarios

For the development of interactive systems for the intuitive use, we suggest a scenario-based design approach to gather requirements and describe the actors in a narrative way. Scenario-based design is a family of techniques in which the use of a future system is concretely described at an early point in the development process [Rosson02]. The involved actors of the visionary scenarios define the users of MIBO.

Building Occupants. The occupants are the main users of the framework and are enabled to use intuitive multimodal commands to retain control over their environment.

Facility Manager. The facility manager of the future must be able to set up desired interaction models in a building. Depending on the layout and presence of particular fixtures, different kinds of interactions are recommended (see section 3.2). The facility manager is able to set up both, user-specific interactions, e.g. for disabled people in a hospital, and uniform interactions for large sets of the users. See section 3.4 for details in customization.

Developer. Developers must easily add new modalities to the framework and combine them flexibly with the existing ones. The domain-specific language *MiboML* to describe the configured interactions for a user or user-group in a space, must provide extension points to integrate new modalities smoothly into the rest of the configuration.

Developer / Usability Engineer. For developers with a strong usability focus or dedicated usability engineers, MIBO must provide a prototyping framework to quickly reconfigure interactions in a space and conduct user studies. The reconfiguration must be

applicable at the runtime of the system and must not require any programming. This is considered as an important property for prototyping and testing of new applications and services, especially for trials and bucket testing [Johnston05]. Eventually, this allows for a highly iterative user-centric design approach.

3.3.1. Scenario 1: Intuitive Building Control

Luzia is employed as an analyst in a bank and works in an innovative office building. Usually she likes to open the blinds next to the desk to enjoy the nice view. Sometimes on very hot days, though, depending on her dress, she prefers to have them closed, so that the room does not heat up too much. She controls the blinds using a multimodal command, saying "Open these blinds" and pointing to the ones next to her desk. When she is still to warm, she uses a voice command "I am hot, make it colder please" to set the HVAC system appropriately. In the evening, especially for reading tasks, she applies gestures to brighten the area around her desk. She uses a gestural command, pointing at the task light and then raising her arm slightly to adjust the specific light level.

3.3.2. Scenario 2: Building Configuration

Heinz is the facility and usability manager of the office building. His job has changed significantly over the past years, when natural user interfaces have been employed in the building. Since more than ten years, he has worked on making the office spaces much more flexible, such that working areas can be arranged flexibly for different projects. If colleagues have to work closely in a group of five people, movable walls can be set up easily to partition the office. As part of these movements, Heinz ensures that the user interface of the office is still usable. For example, if an office gets a second fan or more task lights are installed, a speech command "turn on the fan" would become ambiguous. Therefore, he uses the interactive MIBO editor to set up new interaction definitions for the fan, which implies a point-at gesture to select the fan. These interactions can be configured at runtime by just using drag and drop in the MIBO editor without any programming.

3.3.3. Scenario 3: Prototyping

Simon is a developer with focus on usability engineering. With the enhanced capabilities of the Google Glasses v3.0, he has come up with gaze-tracking as a new modality for controlling fixtures in a building. MIBO allows him to flexibly exchange the gestural *point-at*

commands with *look-at* events and combine them with speech commands. He can put together these modalities in a prototype quickly and conduct user studies. Based on the study results, he can easily change the combination of interactions and modalities with simple configuration, rather than time-consuming custom development. The configurations can be carried out visually using the MIBO editor. This overall simplified process allows for a user-centric iterative development of multimodal systems.

3.3.4. Scenario 4: Ambient Assisted Living

Gertrud is living in an old-people's home and has experienced a stroke. Since then, she cannot speak and not move her left arm anymore. Before the installation of MIBO, Gertrud had to wait for the morning shift of the nurses to finish their briefing, so that they open the blinds for her, even though she has already been awake for two hours. Heinz, the facility manager has then used MIBO to easily configure alternative ways of interaction for Gertrud. Other than the usual controls for patients in the hospital, Gertrud can use her eyes to control fixtures, such as lights, blinds, the TV and heating/cooling devices. While other patients use speech and gestures for their controls, Heinz used the MIBO editor to exchange the "point-at" modality for fixture selection with a "look-at" modality.

3.4. Application Domain Requirements

Based on the previous studies and literature review, functional requirements (FR) and non-functional requirements (NFR) for intuitive controls in smart buildings are derived.

FR1: Multimodal Interaction

To provide natural and intuitive user interfaces, a user must be allowed to interact with buildings in a way similar to human-human communication. This implies the usage of multiple modalities at once, e.g. combining speech, gesture and gaze. It should be optimized for geocentric systems to link spoken references to spatial data, such as fixtures in a room. Additionally, our studies have shown that unimodal inputs can be useful, e.g. for non-spatial commands like setting the temperature (see section 3.2). Thus, unimodal input must be supported as well as multimodal.

Multi-Device Input. As part of multimodality, we also consider the notion of multi-device support. Input devices may include smartphones, watches, glasses but also pervasive devices, such as mounted depth cameras and microphones. These devices are distributed and need to be integrated.

Multi-Dimensional Input. Gestural input spans three spatial dimensions. Additionally, there is the non-spatial acoustic dimension of speech, and both gesture and speech are distributed across the temporal dimension. Related inputs from different modalities have a relation in time which is reported to be between 1000 ms 2000 ms in literature [Afshin11]. We propose a 2000 ms time difference. These temporal and semantic combinations between different input modes have to be taken into account and integrated in a multimodal fusion strategy.

Flexible modality combination. Our user study in section 3.2 has shown that occupants would use very different commands to control their environment, so MIBO has to allow for alternatives. For example, all the following inputs by the user may result in a light turning on:

- Say *Turn on the light*.
- Say *light* and perform an *up* gesture.
- Point at the light and say *turn on*.
- Point at the light and perform an *up* gesture.

Even within one modality, there must be alternatives. While some people might prefer the term *turn on*, some may actually prefer *switch on* or completely different expressions. Wilson and Shafer have pointed out similar requirements to combine modalities flexibly [Wilson03].

FR2: Multi-Device User Feedback

Users have to be provided with meaningful feedback. While some fixtures, such as lights, do provide sufficient feedback about their status, an invisible thermostat requires additional feedback channels. The system has to support feedback on various channels and devices. For example, when occupants use a gestural command with their hands to dim

a light, being recorded by a depth camera, the provision of feedback will inevitably incorporate other devices. These devices may actually be closer to the user, such as a watch. This kind of user feedback is a technical challenge because of the asynchronous, multi-dimensional input, especially being distributed across the temporal dimension. Users in our study (see section 3.2) have accepted a latency of one to two seconds until they repeated their command. However, for some users, speech input and gesture input arrived with a time difference of four seconds. This reveals the challenge of appropriate real-time feedback in multimodal systems.

FR3: Customizability and Multi-User Support

Reeves et al. define guidelines for multimodal user interface design and suggest that multimodal interfaces should adapt to the needs and abilities of different users, as well as different contexts of use [Reeves04]. Individual differences (e.g. age, preferences, skill, sensory or motor impairment) may be captured in a user profile to determine interface settings [Reeves04]. We approach this issue by stressing the requirement to make user interaction customizable for both, different contexts in buildings, different user groups (e.g. cultural differences) and individual users (e.g. disabilities). This customization must be applicable **without programming skills** and **at runtime**.

In an observational study over four years, we could show the importance of *imitation* in the usage of natural user interfaces. When people used HomeGestures – our gesture-based smartphone controller (described in more detail in section 6.1), we surveyed 17 users and also asked how they learned to use the gestures. Measured on a likert-scale [Likert32], 75% of 16 users who answered the question, agreed or strongly agreed that they learned how to use HomeGestures by observing when another person gestured with it. While other research proposes user-customized interactions [Reeves04] [Lou13] [Liu09], our study showed the importance of imitation in natural user interfaces. If every single user sets up own styles of gestures, voice commands and combinations, the effect of imitation and learnability might become problematic. We therefore suggest — whenever possible -– a unified set of interactions for a certain space and larger groups of users. We propose customization on three dimensions:

- **Cultural customization.** Since multimodal user interaction aims at making human-machine interfaces more similar to the interaction between humans, we have to consider differences also among cultures. This becomes not only clear in the different

languages and expressions but also in the usage of gestures. For example, while Europeans interpret nodding as a form of agreement, Indians would interpret the same as denial. MIBO must enable simple customization of interaction models in buildings for different cultures.

- **Building customization.** Buildings, including their interior, can be very different from each other. Our experiments have shown a significant relation to the involved interactions. While a visible thermostat might have a spatial reference and therefore gestures might be used to refer to it, an invisible thermostat could hardly be referred to and is more suitable to be controlled by voice. Also, the presence of multiple fixtures of the same type, make it hard for users to refer to the correct fixture with just voice. However, if there is only one fixture of that type, a command such as *"turn on the fan"* might be a comfortable and appropriate way to control ventilation. MIBO must allow simple customization of interactions in buildings, whenever the layout of a space is created or rearranged.

- **User customization.** As introduced in section 2.3.4, about 15% of the world's population have some form of disability. Therefore, MIBO must support user-specific customization to overcome individual limitations or preferences in the usage of particular modalities. Features like automatic face and voice recognition might be incorporated to realize automatic adaption.

To realize customization, we propose an object-oriented approach, wherein particular buildings, user groups and individual users can inherit and override interactions.

FR4: Context Awareness

Our user study also showed the importance of context awareness. For example, people wanted to be able to refer to the same kinds of fixtures that they controlled recently (e.g. *"turn on fan"*, then *"turn it off again"*). Additionally, the incorporation of user-specific context information, such as *"turn on the fan in my office"* is required. The consideration of context eventually allows the transition from an *instrumented* environment to a truly *smart* environment.

NFR1: Robustness

Multimodal input, including gestures, speech, gaze-tracking or even brain-computer input has to be expected to be ambiguous, imprecise and incomplete. This fuzzy input has to be integrated in a multimodal fusion strategy. MIBO must be robust to this fuzzy input in a non-destructive way, i.e. fuzzy input must not break the expected system behavior. Furthermore, MIBO must allow for a precise error prevention, error correction, error handling and error recovery.

NFR2: Performance / Real-Time

To gain user acceptance for a multimodal system, response times must be reasonably small. Our own wizard-of-oz study, as well as related work by Flippo et al. [Flippo03] has shown that if the system takes too long to process a user's command, the user will think the command was not understood and repeat it, resulting in confusion when the command gets carried out twice. Flippo states that this circumstance results in annoyance and should be avoided at all cost and proposes an overall response time below one second [Flippo03]. Our studies have shown that users tend to repeat not-recognized commands after one to two seconds.

This leads to the challenging requirement of real-time processing of multimodal input, including appropriate user feedback. We base MIBO on *soft* real-time requirements, as categorized by Shin and Ramanathan [Shin94]. This means that the usefulness of a result degrades after its deadline and thus significantly degrades the system's quality of service. *Firm* or *hard* real-time requirements might be relevant in buildings, especially for safety-relevant systems. However, they are out of the scope of this research.

The required real-time interpretation and manipulation of data is an important difference to other related work [Bolt80] [Johnston97] [Oviatt97]. In order to provide appropriate feedback to occupants, the multimodal input for controlling building fixtures must execute actions before the whole command is actually completed. For example, our studies have shown that occupants would like to see a light dimming while they are using gestures to control it. However, the gesture is not yet fully completed by that time. MIBO must efficiently manage the parallel input streams as well as performing the recognition and fusion in the presence of temporal constraints. This makes it necessary to process input incrementally (see [Afshin11]) and also requires a more lightweight system to comply to the performance constraints.

3.5. Solution Domain Requirements

In order to consider the developer as future user of the MIBO framework, extensibility has to be taken into account as a major solution domain requirement.

NFR3: Extensibility of Supported Modalities

To support the developers of multimodal systems in buildings, new emerging modalities, such as gaze-tracking and brain-computer interfaces have to be integratable using adapters, without changes in the system design. *MiboML*, which is the multimodal definition syntax, to define the interaction models has to be extensible to this respect as well.

NFR4: Device Independence

Because of the heterogeneity of buildings and an ever-changing market of addressable fixtures with various communication protocols, MIBO has to be independent from all underlying technical protocols and device specifics. Developers have to be provided with unification and standardization of user intention recognition regardless of the employed technology. MIBO has to be extensible to incorporate new device protocols.

NFR5: Legacy Support

It is estimated that less than 10% of North American buildings is younger than 25 years [O'Connor04]. Therefore, the support of legacy systems is an important requirement to the incorporation of innovative control systems in buildings.

3.6. Summary

Based on user studies, literature and visionary scenarios, we identified requirements and actors for MIBO and refined these throughout the iterative creation process of the framework. The particular scope of functional and non-functional requirements is based on the domain of controls in smart buildings. The next chapter approaches these requirements by presenting the concepts of an object-oriented and pattern-based framework design.

4. Framework Design

This chapter describes the design of the MIBO framework which realizes the requirements for multimodal intuitive building controls. It starts with an introduction in form of a comprehensive review of related work on multimodal parsing, integration and related architectural approaches. An analysis model is then created to formalize the information and objects that exist in the application domain of multimodal interaction in smart buildings. Afterwards, the reference architecture of MIBO is presented, as well as concepts for realizing multimodal user feedback.

4.1. Introduction

Since Bolt presented "Put-that-there" [Bolt80], one of the first multimodal systems, Johnston et al. state that in the sixteen years after its publication, research on "multimodal integration has not yielded a reusable scalable architecture for the construction of multimodal systems" [Johnston97]. At the end of the nineties, Johnston initiated a phase with substantial research and major contributions in multimodal interactions. However, while the idea of rapid development of these interfaces has been a driver for a number of contributions, e. g. [Johnston98b], [Johnston09a], [Afshin11], none of these frameworks were tailored to the specific needs in smart buildings, covering all the specific non-functional requirements, elicited in section 3.4. In particular, none of them proved to have a simple approach that allows end-users and facility managers of buildings to set up the required interaction model for multimodal building controls and adjust it to the occupant's needs. At the same time, the support for continuous interactions through iterative multimodal fusion with immediate user feedback was not covered by most of the related work.

4.1.1. Multimodal Parsing and Integration

The first approach to multimodal integration of spoken and gestural input was formalized by Johnston et al. [Johnston97]. It was driven by a unification operation over typed fea-

ture structures representing the semantic contributions of the various input modes. His presented approach overcomes the limitations of previous work by allowing for a broader range of input beyond simple deictic pointing gestures. Unlike speech-driven approaches (c.f. [Bolt80], [Neal91], [Koons93], [Wauchope94]), it is fully multimodal in that all elements of a command can be in either mode [Johnston98b]. The notion of *Typed Feature Structures* [Carpenter92] has been introduced by Carpenter. They are essentially representing partial information expressed as attributes and values. In a graph-based representation, each value of a feature is either undefined or another feature structure. Unification can be done among these by determining the consistency of two partial information and combining them to a single result. Carpenter enhanced feature structures with object-oriented types, including multiple inheritance and admissible types for values. Johnston proposes the usage of *Typed Feature Structures* and their unification to determine whether a given piece of gestural input is compatible with a given piece of spoken input and, if they are compatible, to combine the two inputs into a single result that can be interpreted by the system. Figure 4.1 shows the feature structure for an example command [Johnston97].

$$
\begin{bmatrix}
\textbf{cat} : \mathit{subcat\_command} \\
\textbf{content} : \begin{bmatrix}
\textbf{fsTYPE} : \mathit{create\_line} \\
\textbf{object} : \begin{bmatrix} \textbf{fsTYPE} : \mathit{wall\_obj} \\ \textbf{style} : \mathit{sand\_bag} \\ \textbf{color} : \mathit{grey} \end{bmatrix} \\
\textbf{location} : \begin{bmatrix} \textbf{fsTYPE} : \mathit{line} \\ \textbf{coordlist} : [[1],[2]] \end{bmatrix}
\end{bmatrix} \\
\textbf{time} : [3] \\
\textbf{subcat} : \begin{bmatrix}
\textbf{first} : \begin{bmatrix} \textbf{cat} : \mathit{spatial\_gesture} \\ \textbf{content} : \begin{bmatrix} \textbf{fsTYPE} : \mathit{point} \\ \textbf{coord} : [1] \end{bmatrix} \\ \textbf{time} : [4] \end{bmatrix} \\
\textbf{constraints} : [\mathit{overlap}([3],[4]) \vee \mathit{follow}([3],[4],4)] \\
\textbf{rest} : \begin{bmatrix}
\textbf{first} : \begin{bmatrix} \textbf{cat} : \mathit{spatial\_gesture} \\ \textbf{content} : \begin{bmatrix} \textbf{fsTYPE} : \mathit{point} \\ \textbf{coord} : [2] \end{bmatrix} \\ \textbf{time} : [5] \end{bmatrix} \\
\textbf{constraints} : [\mathit{follow}([5],[4],5)] \\
\textbf{rest} : \mathit{end}
\end{bmatrix}
\end{bmatrix}
\end{bmatrix}
$$

Figure 4.1.: Example feature structure for a "Sandbag wall from here to here" command. Source: [Johnston97].

Johnston later writes that his own approach faces fundamental architectural problems [Johnston98b], mainly because the integration strategy is hard-coded and the input structures support only multimodal combinations consisting of a single spoken element and a single gesture [Johnston98b]. The approach was improved in a number of publications, e. g., enhancing it with declarative integration strategies that are interpreted by an incre-

mental multidimensional parser [Johnston98b] [Johnston98a]. Johnston later showed how the structure and interpretation of multimodal commands can be captured declaratively in a multimodal context-free grammar, being interpreted by a weighted finite state automaton [Johnston00] [Johnston05]. However, their grammar is intended as a syntactical formalization of allowed multimodal commands. Unlike MIBO, it does not allow the representation of a full interaction model, including the actual semantics of a multimodal command.

Sun et al. show an approach for multi-sensory data fusion in multimodal systems which is also based on unification-based grammars and typed feature structures [Sun06]. However, the multimodal integration strategy is strictly linear and implies tight-coupling to the modalities which makes it less flexible and extensible. Since MIBO is designed as a framework to ease the development of multimodal interactions in buildings even with yet unknown modalities, the modalities must be loosely coupled to the framework to allow for an extensible and flexible approach.

Flippo et al. propose a framework for rapid development of multimodal interfaces to overcome the marginal penetration of multimodal applications in the market [Flippo03] . They show that a framework for multimodal interaction can actually reduce the development effort for future applications. We consider their approach for end-user customization as not suitable for real world building occupants because of their programming-like models. Additionally, there is no native support for multimodal interactions with continuous outputs, such as up/down gestures to control lights which is an essential requirement in the domain of smart buildings.

4.1.2. Architectures and Frameworks for Multimodal Interfaces

The *Open Agent Architecture* presented by Cohen et al. is designed to support distributed execution of user requests, interoperability of multiple application subsystems and transparent delegation, i.e. users should not need to know where their requests are being executed [Cohen94]. The Open Agent Architecture is a blackboard-based framework wherein the blackboard acts as a server process with several client agents. Figure 4.2 shows an example of the agent interaction with several agents acting on a shared blackboard. Prolog is used as the interagent communication language [Cohen94]. Agent Architectures have been commonly adopted for multimodal systems [Johnston97] [Rieger05] [Reithinger03]. A comparison of MIBO to the *Open Agent Architecture* is conducted in section 4.3.1.

Fernandes et al. describe a middleware framework to manage multimodal applications

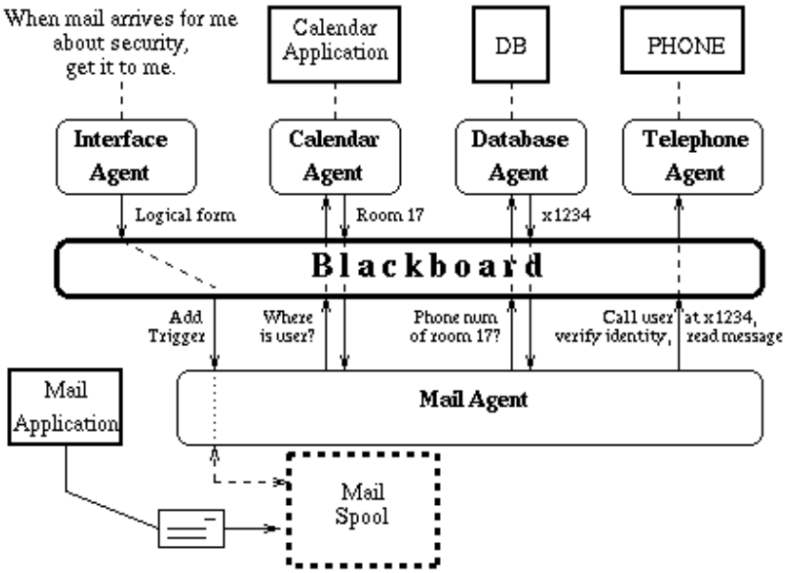


Figure 4.2.: Example of agent interaction in the Open Agent Architecture. Source: [Cohen94].

and interfaces in a reusable and extensible manner [Fernandes07]. According to the authors, the main goal of the framework is "to manage input modalities and applications separately allowing that each component can be reused and extended" [Fernandes07]. The input modalities offer reusable capability tokens to applications, such as mouse commands, gesture tokens, speech commands. Delivering the same tokens from several input sources allows a flexible wiring between the modalities and applications. The architecture is based on a message-oriented approach. The authors focus on the domain of collaborative environments and demonstrate the system in the context of interaction with a tiled display wall. However, the parsing of *composed inputs* incorporating more than one modality at the same time remains to future work. The differentiation between user(-groups) and the incorporation of user context is also out of the scope of Fernandes' work [Fernandes07]. Both aspects are supported by the MIBO framework.

In the area of *Ambient Assisted Living*, D'Andrea et al. show a framework to process information derived from multiple input modalities, model these inputs in an appropriate representation and integrate them into a joint semantic interpretation [D'Andrea09]. The scope of the work is limited to the domain of *Ambient Assisted Living* which is only a subdomain of smart buildings. MIBO targets multimodal interactions in a broader context of

smart buildings in general.

In the area of multimodal dialogues, *SmartKom* combines speech and natural gestures to provide a homogeneous and pleasing interaction experience for conversational dialogues [Reithinger03]. The goal of SmartKom is to provide a generic processing approach covering a whole variety of applications and modalities with a uniform backbone system. SmartKom also includes an XML-based markup language *M3L* to provide a format for the data interfaces within the complex dialogue system [Reithinger03]. However, SmartKom's focus is on the support of a dialogue system. The end-user customization of interaction models and support for continuous interactions, which is a major requirement to MIBO, is not described by Reithinger et al. [Reithinger03].

As part of the *OpenInterface Project*, a European Specific Targeted Research Project, the OpenInterface-platform is presented [Lawson09a]. It is an open-source software solution, designed to support fast prototyping and implementation of interactive multimodal systems. It aims at enabling the reuse of existing components and iteratively designing multimodal systems with minimal programming effort [Lawson09a]. Although a graphical editor allows rapid prototyping, this editor is still not feasible for end-users but developer experts. MIBO provides a graphical user interface which enables occupants and facility managers to customize their interaction model for controlling fixtures in buildings.

The World Wide Web Consortium (W3C) has specified a multimodal framework for extending the web to support multiple modes of interaction [Larson09]. It proposes a set of properties and standards that are at a level of abstraction above an architecture. It has also been extended with the *Extensible Multimodal Annotation Markup Language (EMMA)* as a representation language for inputs to multimodal systems [Johnston09b]. Johnston has illustrated the capabilities of the EMMA standard through examination of its use in a series of multimodal applications [Johnston05]. Feuerstack et al. have shown a similar approach called *MINT* which is a platform that enables designing and running multimodal web applications [Feuerstack12]. However, the W3C's scope is limited to applications in the web, while MIBO targets interactions in buildings.

Lo et al. present a framework to integrate interface devices of multiple modalities [Lo10]. The framework provides a set of libraries for developers to create their own multimodal applications without caring about the low-level details of each device. Notably, it also includes an interaction grammar that generalizes events from different input devices and supports the translation of events into commands. However, their approach to provide developers with the ability to create the presented data flow is too simple because it does

not allow fusion of multiple arbitrary modalities and the composition of multiple events to a higher level command which is a major requirement to MIBO.

Afshin et al. focus on enabling real-time incremental processing of the multimodal input to give immediate feedback to the user [Afshin11]. This is important in multimodal systems to improve the response times of the system. If the latency is too high, it may lead to repetition of commands from the user, more ambiguity in the recognition process and user annoyance [Flippo03]. The authors designed the framework to perform natural language understanding, multimodal integration and semantic analysis with an incremental pipeline [Afshin11]. It also includes a multimodal grammar language for the semantic analysis representation. The framework detects completely parsed hypotheses even when more information is still to be added and fires events in such cases for immediate user feedback. This flexibility, however, is traded off by a lack in customizability for end-users since their approach requires programming skills in Prolog, while MIBO allows customization even for end-users.

Hoste et al. have shown a framework that unifies the multimodal fusion across different levels of abstractions [Hoste11]. They argue that existing multimodal interaction frameworks excel at one specific fusion level but encounter major difficulties at other levels. They present an architecture which makes information from all levels available on a common fact base in contrast to a traditional chain-based solution [Hoste11]. However, this results in performance issues that are not acceptable for continuous interactions in smart buildings. Furthermore, the lack of conflict resolution decreases maintainability.

MIBO applies multimodal integration for controlling smart buildings. This means, that multimodal input finally results in commands that are executed by a building control system. These systems are introduced in the following section.

4.1.3. Building Automation and Control Systems

The integration of building control systems using standard protocols has been a continuous, though unsuccessful endeavor. The multitude of available fieldbus technologies (e.g. KNX/EIB, LON, DALI, CAN, BACnet, ModBus, M-Bus, ZigBee, Zwave) as well as all kinds of proprietary protocols, reveal that a consistent communication standard has not yet been found. The above mentioned fieldbus technologies continue to use parallel networks that do not collaborate. Hersent concludes that this is a major reason for the success of the Internet of Things in its current form [Hersent12]. Aldrich claims that because of the

lack of a common protocol, the smart building industry has tended to focus on simple on-off switching systems for applications that do not require additional network installation [Aldrich03].

The ISO Open Systems Interconnection (OSI) model can be used to bring order into the amount of different protocols and standards. In the following, we describe a number of popular protocols and standards using this model, shown in figure 4.3.

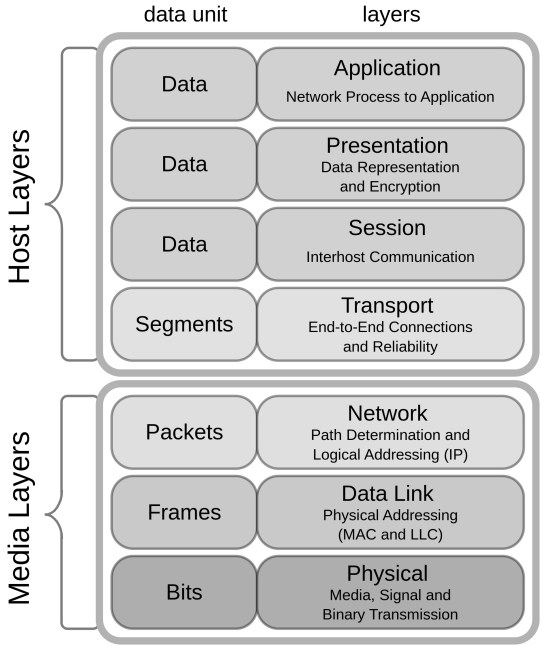


Figure 4.3.: The ISO Open Systems Interconnection (OSI) model.

Media Layers. Transmission and reception of information from the physical environment over a physical medium take place in the media layers. Sensor data is collected and transformed into a representation suitable for reliable transmission and processing over the network. The media layers also represent the underlying network topology. The **European Installation Bus** (EIB) and its successor **KNX** have been specified on this level in the norm ISO/IEC 14543. The standard allows for serial communication to enable sending coded signals over twisted pair wires. KNX devices provide hard-coded addresses, similar to MAC addresses in IEEE 802 network standards. The KNX protocol defines specific control frames for switching, dimming and setting of parameters on the application

layer. The standard was not motivated by today's requirements in ubiquitous computing but rather to reduce the overall current-carrying wiring in buildings, e.g. for simple multiway switching. **LonWorks** (LON) specified in ANSI/CEA-709.1, has turned out to be a rival protocol with a stronger focus on lighting and HVAC. Other standards, such as **ZigBee** and **Bluetooth**, provide wireless media layers but they have turned out to contain too much overhead for lightweight communication using radio frequency (RF). Therefore, **EnOcean** has been specified in ISO/IEC 14543-3-10 to allow for energy harvesting using ultra low power electronics and lightweight RF communication. This enables wireless communications between batteryless sensors, switches, controllers and gateways.

Host Layers. The host layers, and in particular the application layer, enable building automation, including autonomously executed sequences. These layers operate on data prepared by the media layers. The **BACnet** (Building Automation and Control Networks) communication protocol, specified in ISO 16484-5 includes functionality for device discovery and data sharing services, such as Read-Property and Write-Property. BACnet supports analog inputs/outputs, binary inputs/outputs, multi-state inputs/outputs, calendar consideration, events, files, notifications and others. The protocol can work on various types of media layers, including Ethernet, KNX and LON. Other protocols, such as the **OPC Unified Architecture**, developed by the OPC Foundation, are operating on the same level and are usually aiming at not only transporting information but also providing machine-readable semantics on the application layer.

Broker Architectures

The multitude of protocols and requirements for interoperability have made it necessary to integrate several protocols. Therefore, brokers decouple these systems from each other, using the broker architectural pattern, formalized by Buschmann et al. [Buschmann96]. The broker can be described as a middleware layer [Augusto10b] between applications and services on the one side and sensors/actuators on the other side (see figure 4.4).

For such purpose, Nosovic, Peters et al. have described an extensible broker for controlling instrumented and smart environments [Nosovic14]. Their system design is based on applications and services communicating over an Enterprise Service Bus (ESB). It provides communication protocol abstractions that enable the broker to be extended with arbitrary protocol implementations to communicate with sensors and actuators. Applications and

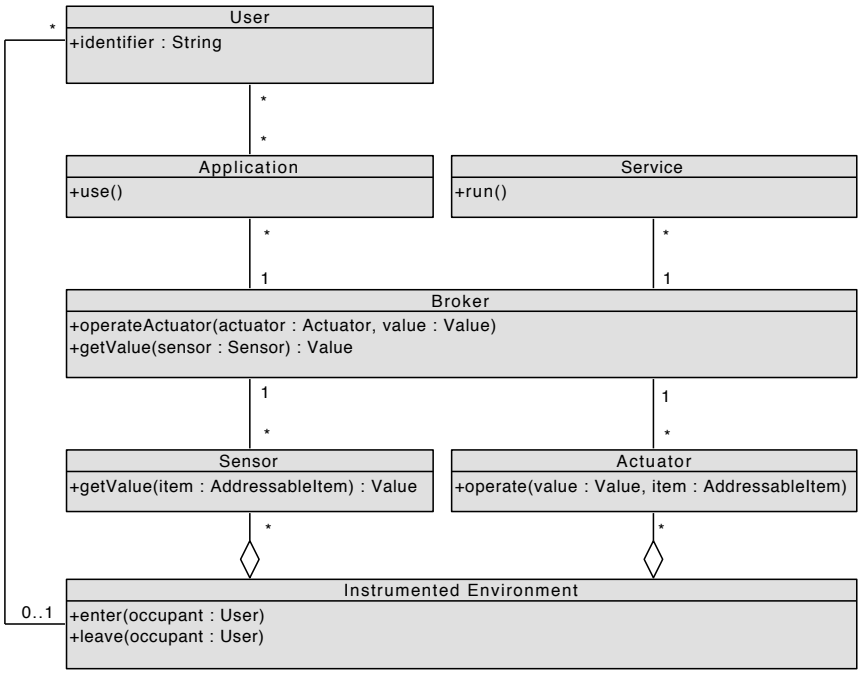


Figure 4.4.: Broker architecture for the abstraction of applications and hardware.

services can operate the broker using a uniform facade [Nosovic14]. Similar approaches using a broker as intermediate layer have been shown by Peters [Peters11] and Koß et al. [Koß12]. Afterwards, open-source solutions, such as *OpenHAB* [1] have become available with similar functionality to abstract applications and services from the protocol implementations of sensors and actuators.

4.2. Application Domain

Based on the requirements elicitation in chapter 3, an analysis model is created to formalize the information and objects that exist in the application domain of multimodal interaction in smart buildings.
The analysis object model in figure 4.5 shows four interaction modalities:

- **Gesture.** Describes the interaction of a user, using gestures with the arms and hands.
- **Voice.** Describes the interaction of a user, using spoken natural language.

[1] http://www.openhab.org

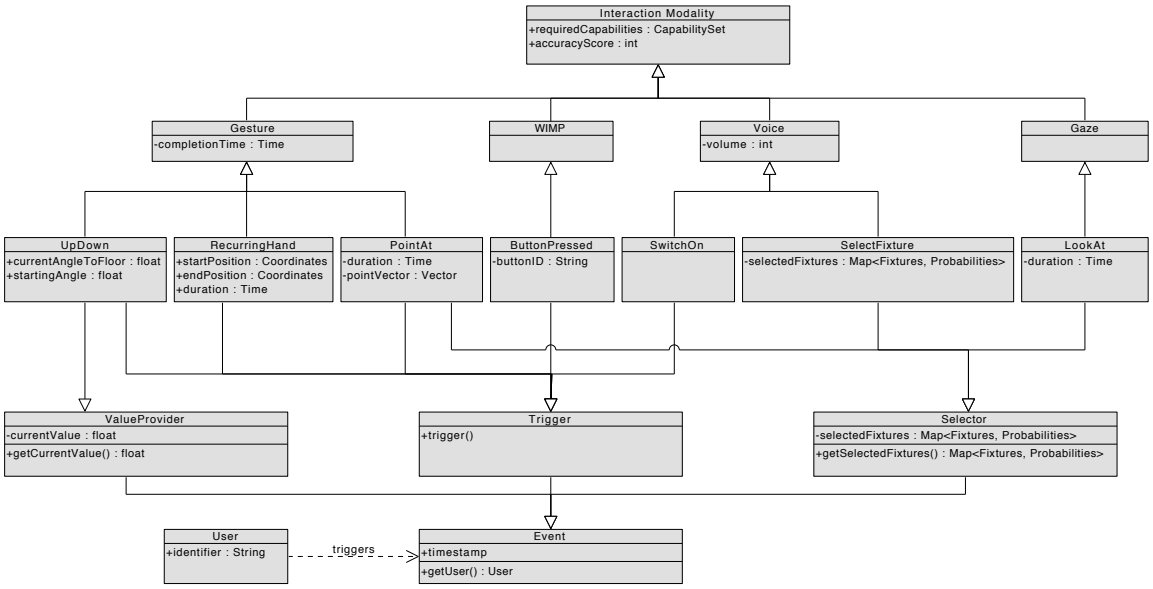


Figure 4.5.: Analysis Object Model of interaction modalities and exemplary associated event categories (UML class diagram).

- **Gaze.** Describes the interaction of a user, using the gaze of the eyes.
- **WIMP.** Describes the interaction of a user, using a traditional interface, e.g. incorporating a button in hardware or software. WIMP is the common abbreviation for Windows-Icons-Menus-Pointers and is the foundation of graphical user interfaces (GUIs).

This list of modalities is not exhaustive and extensibility is one of the key requirements of MIBO. The same applies for the concrete events, such as up/down gestures, point-at gestures, look-at gazes or switch-on voice events. These events are the only non-abstract classes of the model and use multiple inheritance because they represent modalities on the one hand and event categories on the other. Each event is a subclass of one or more of the categories `Trigger`, `Selector` and `ValueProvider`.

- **Trigger.** A `Trigger` describes a discrete event that is able to cause a particular action. For instance, a switch-on voice event may be used to trigger an arbitrary fixture.
- **ValueProvider.** A `ValueProvider` is an event that provides a value when it is triggered. A `ValueProvider` may also be used to provide a value over a continuous

duration. For instance, an up/down gesture event may be used to continuously adjust the light level of a dimmable light.

- **Selectors.** A `Selector` is an event that selects a particular fixture to take action. For instance, a point-at gesture can be used to select one or several particular fixtures to control their value.

It is noteworthy that events may inherit from multiple parent classes at once. For example, a point-at gesture may be used to select a fixture but also just to trigger an action by pointing for a specific minimum duration. In practice, it is up to the developer of new events, to determine its categorization in the application domain and then implement the required interfaces.

The model in figure 4.6 shows how `Definition` objects can be used to describe an interaction model for smart buildings.

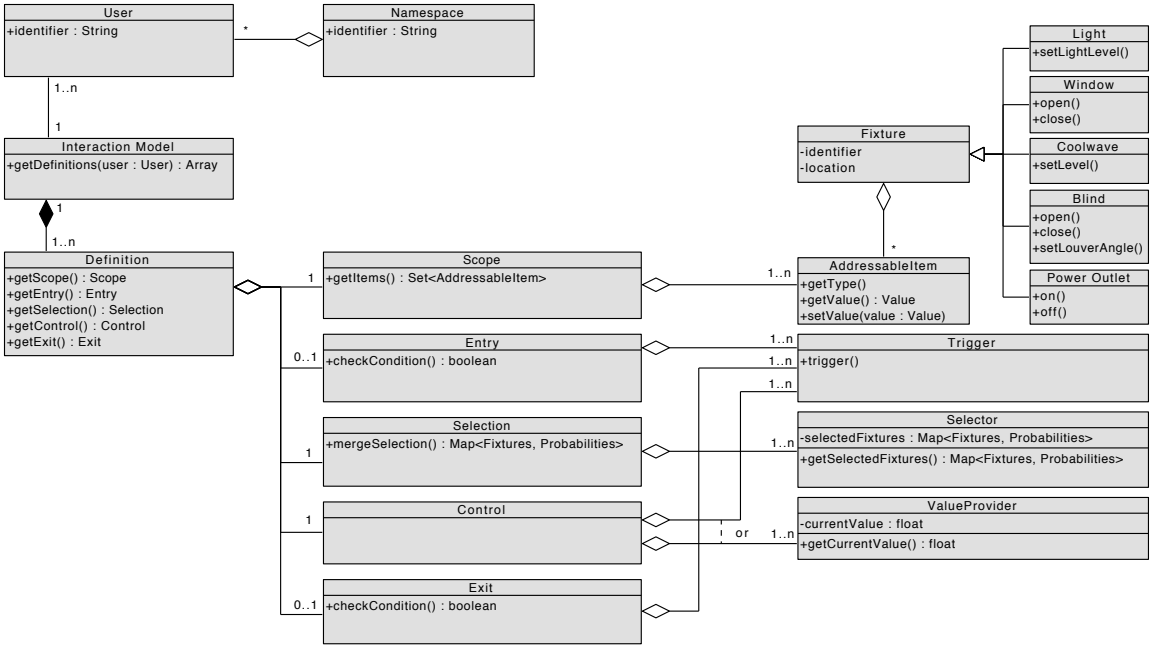


Figure 4.6.: Analysis Object Model for the definition of multimodal interactions to control building fixtures (UML class diagram).

Scope. A definition consists of a `Scope`, specifying which fixtures are targeted by this definition. The scope may explicitly include or explicitly exclude particular fixtures or fix-

ture types. For example, a definition might define to use an up/down gesture to raise/lower blinds but not influencing motorized windows (which may reside almost at the same physical location in this case). Additionally, a fixture consists of arbitrary addressable items. For instance, blinds may have two addressable items, that is, the motor that raises/lowers the blinds and the motor that adjusts the angle of the slats to redirect the light.

Entry. A definition may have an entry condition that must be met to start an interaction. For example, a voice command *"Start Dimming"* might be used to start a dimming process which is then performed by gestures. Any event may not only origin from arbitrary modalities but also from arbitrary devices. For example, a button on a wearable computing device, such as a watch, might be used to trigger the entry condition and start a recognition process in another modality (e.g. voice or gesture). Several triggers can be combined to form an entry condition.

Selection. A definition must always have at least one `Selector` that specifies the fixtures that the user is about to control. Selectors may be marked as optional if it is optional for the user to provide additional selectors. For instance, a user may point at a fixture and say either *"turn on"* or *"turn on that light"*. While the first alternative only contains one selector (point-at gesture), the latter alternative provides two selectors (point-at gesture and voice fixture type selector "that light"). Multiple selectors usually remove ambiguity because the user might have pointed not exactly at the desk light but also into the direction of the fan. By adding the information that the selected fixture is a *light*, the overall fusion result is improved. However, adding more specific information may also be more circumstantial for the user.

Control. The control of a definition determines the appropriate combination of `Trigger` and `ValueProvider` that operate the addressable items of a fixture. For instance, a trigger may increase the value of an item or set it to a specific value.

Exit. Analogous to the entry condition, a definition may also have an exit condition that exits the control process immediately when it is fulfilled. For example, a voice command *"OK"* may be used to acknowledge and end a dimming process which has been performed by a gesture. Several triggers can be combined to form an exit condition.

Each definition is part of an interaction model that may be applied for individual users, as well as for groups of users or entire namespaces. The following example in figure 4.7 shows a definition to dim a light using gestures and voice and depicts the categorization of events:

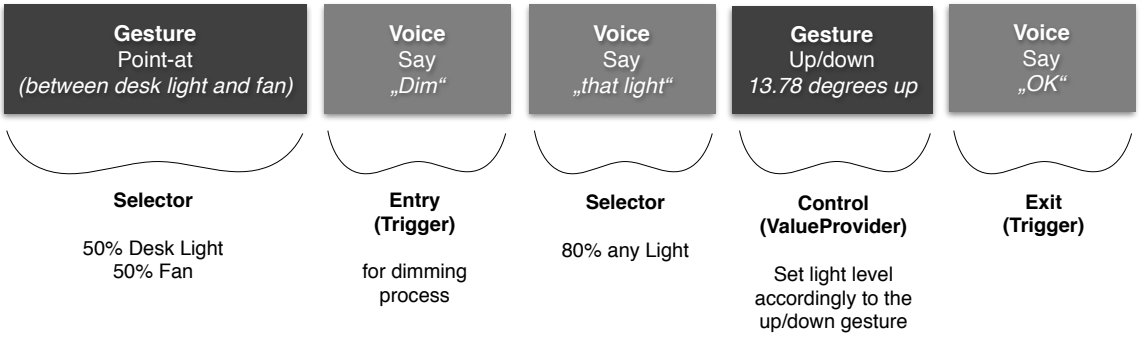


Figure 4.7.: Example for a control to dim a light using gestures and voice.

While the point-at gesture returns a selection probability of 50% for the desk light and 50% for the fan, the additional selector "light" determines with a confidence of 80% that the selection should be some kind of light. The voice event *"Dim"* actually triggers the dimming process as entry condition. A subsequent up/down gesture with the user's arm can be used to set the light level in a fine granular way. The voice event *"OK"* finally exits the dimming process, so that the arm movement is no longer used to dim the light.

4.3. MIBO Software Architecture

The multimodal research community has adopted a number of architectural styles for information processing. One common approach involves *multi-agent architectures*, such as the Open Agent Architecture [Cohen94] and Adaptive Agent Architecture [Kumar00]. A multi-agent architecture is used commonly in multimodal systems, whereas the QuickSet system [Cohen97] was the first known. MIBO's overall software architecture is based on agents for each modality and the blackboard pattern to resolve a unification-based grammar. This approach separates three parts of fusion. First, it obtains and keeps data from the modality agents, then fuses that data to come to an unambiguous meaning, and finally calls application code to take an action based on that meaning and the particular configuration of MIBO using the interaction model, described in MiboML. The separation of

these tasks makes the framework more flexible and therefore applicable to a wider range of applications and modalities.

4.3.1. Blackboard Architectural Style

The blackboard pattern describes a solution approach analogous to human experts gathering around a blackboard to solve a problem. It has been first demonstrated in the Hearsay II system [Erman80] by Erman, Reddy et al. for speech recognition and later formulated as an architectural pattern by Buschmann et al. [Buschmann96]. *Knowledge sources* are working together as self-activating, asynchronous, parallel processes on a problem without a predetermined algorithmic solution. The problem is partitioned into related subproblems and interacting reasoning techniques are used to solve the component problems, with an evolving solution obtained by combining the results for each subproblem. The blackboard is the repository for the problem and partial solutions, called *hypotheses*. The representation of knowledge as hypotheses suits well to the requirements of multimodal input fusion. Additionally, since the used modalities are only known at runtime and yet unknown modalities may be added in the future, the framework design has to be extensible and maintainable. Therefore, MIBO's blackboard approach provides a modular problem solving process which does not follow a strict sequence. In contrast to input from traditional GUIs, being generally deterministic, multimodal input streams with gestures and speech have to be interpreted by probabilistic recognizers [Dumas09] and thus, hypotheses are weighted by a degree of uncertainty. This underlines the design goals of fault tolerance and robustness which can be approached using the blackboard pattern.
The component diagram in figure 4.8 shows the blackboard as the overall architectural style and central repository for all knowledge exchange. This decouples the modalities, the MIBO definition evaluation, as well as the command execution with clear interfaces and a strong need-to-know principle. The knowledge sources act independently from each other, accessing basic functionality from the common blackboard component. The approach fosters the use of software engineering principles such as modularization to support a growing set of input modalities as well as to enable the flexible definition and configuration of interaction models. Compared to other blackboard architectures, MIBO's multimodal input streams using gestures and speech events are not always clearly temporally delimited and thus, require continuous tracking, incremental interpretation and command execution. For example, a user who dims a light using an up/down-gesture would like to see the effect of the dimming instantaneously, even before completing the

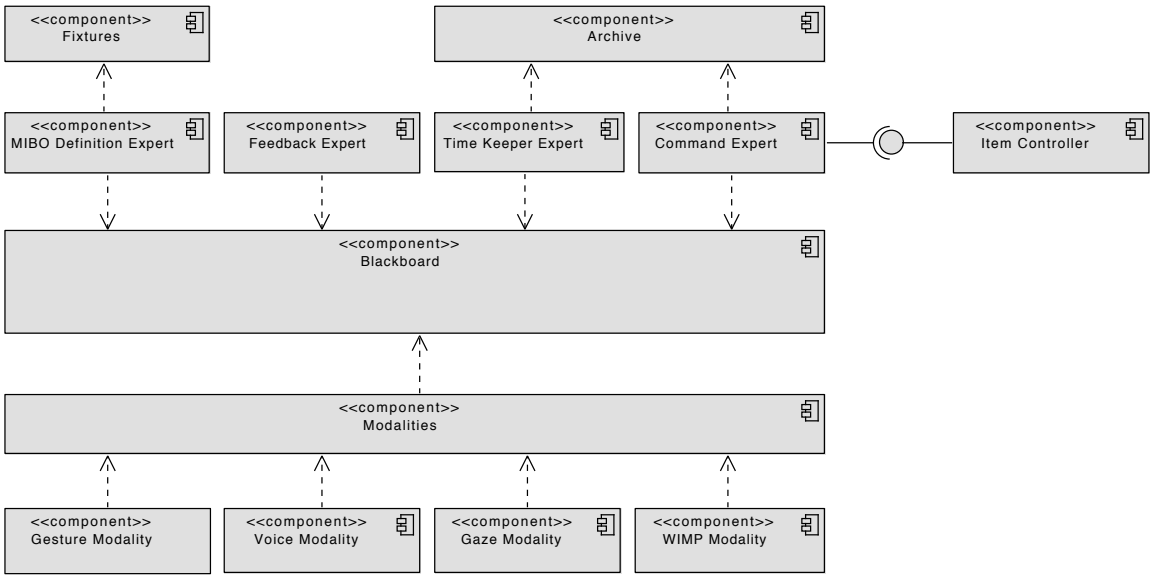


Figure 4.8.: Components of MIBO as UML component diagram.

whole gesture. On the other hand, the blackboard performs temporal integration over the compositions of each command by providing a memory of the current components, such as the entry condition, selector and control request. This makes it unnecessary to specify these components at exactly the same moment in time. For instance, a user might want to point at a light and then say *"turn on"*. The latter hypothesis may arrive several seconds after the pointing. On the other hand, the user might prefer to start with the voice command *"turn on"* and then point at one or several other fixtures in a specific time frame. Our architecture allows every input event to define its own default validity time on the blackboard. For example, our case studies in chapter 6 have shown that a point-at gesture should remain on the blackboard for four seconds if no applicable interaction is triggered during that time frame.

In order to abstract from the particular modalities, MIBO uses an object-oriented approach in which the hypotheses can contain potentially any event that is defined by a modality. However, the implementation of the modalities must ensure that the interfaces of these hypotheses are meaningful to the knowledge sources of the particular modality. During the reasoning process of the blackboard, events are eventually generalized into `Trigger`, `Selector` or `ValueProvider` (see section 4.2). This allows the problem solving to be independent from specific modalities once a certain solution level has

been reached. Beside the beneficial usage of the blackboard for problem solving support, changeability, maintainability, fault tolerance and robustness, the blackboard also entails two major drawbacks:

1. **Difficulty of testing.** It is impossible to reproduce every combination of non-deterministic multimodal input. This leads to difficulties in testing. While test cases with a specified particular input may work correctly, additional random data on the blackboard may influence the final outcome. We therefore recommend to build test cases that are repeated with a large number of additional combinations of unrelated hypotheses on the blackboard to reduce the risk of side effects.

2. **High development effort.** Blackboard systems often evolve over years. However, MIBO has been created to take away this burden from developers of multimodal interactions in buildings, reducing the development effort for other developers.

Comparison to Open Agent Architecture

The Open Agent Architecture was presented by Cohen [Cohen94] and proposed by Johnston [Johnston98a] for multimodal fusion. Like in Johnston's approach, MIBO uses agents to represent the modalities. Thereby, the blackboard subsystem serves as a server, while the modality agents are distributed throughout various clients. The agents have a local proxy knowledge source at the blackboard but the actual input data is provided by the distributed agents which may be written in arbitrary programming languages. Different to the Open Agent Architecture, the agents do not put requests for information on the blackboard and do not communicate via Prolog. Instead, the agents continuously put all available knowledge on the blackboard. They may consist of multiple knowledge sources to separate event collection, parsing and pattern recognition (e.g. to recognize particular movements as a gesture). The multimodal integration is performed by higher level reasoning. The *MIBO Definition Expert* continuously analyzes the multimodal input on the highest level to create appropriate executable commands, according to the interaction model.

Finally, the Open Agent Architecture is limited to an architectural style and does not consider the application domain of multimodal integration and smart building controls. MIBO covers the specific requirements of this domain using the concepts for multimodal fusion and the definition and separation of interaction models for smart building controls.

4.3.2. Subsystem Decomposition

The following section describes the key concepts of the subsystem decomposition and refines solution objects of MIBO's software architecture to realize the defined subsystems. Thereby, the boundaries between objects become visible and show MIBO's clear separation of concerns and low coupling between the interaction modalities. Figure 4.9 shows the static structure of the framework in a UML class diagram.

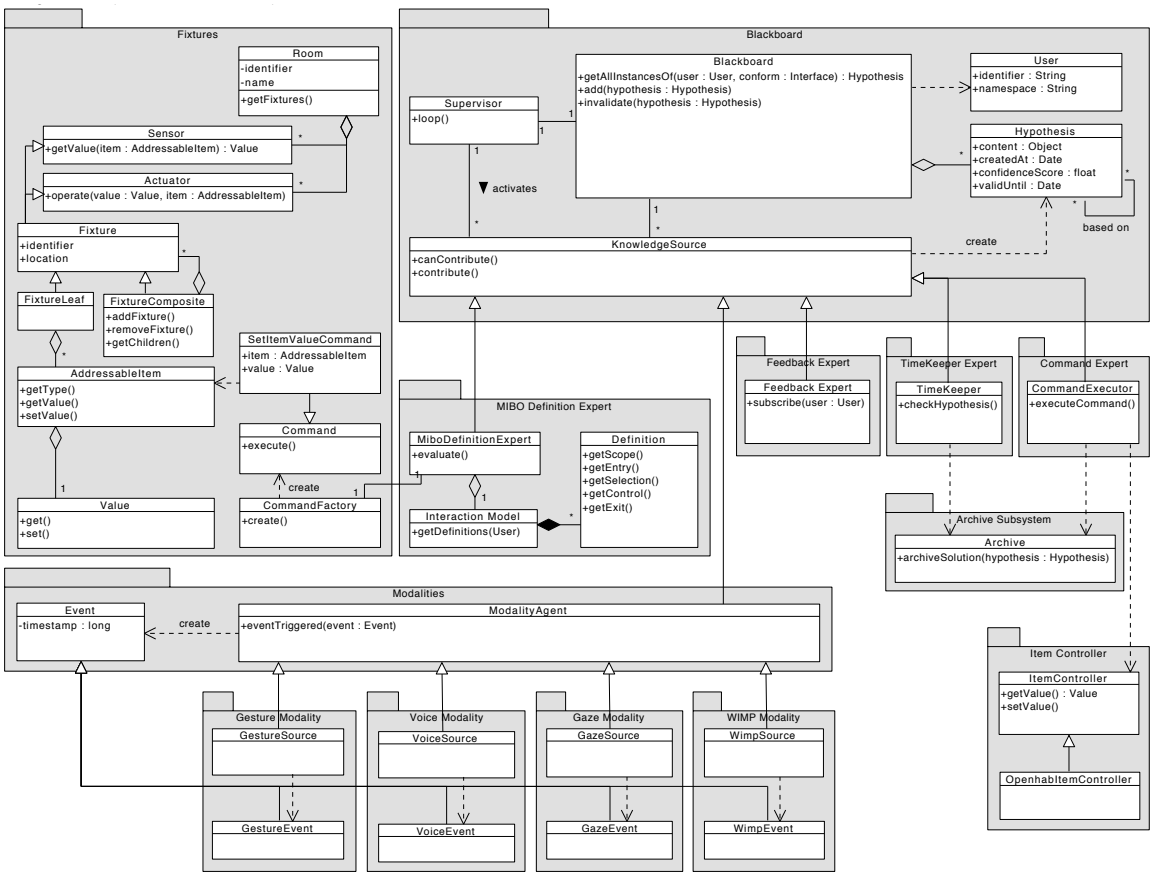


Figure 4.9.: Subsystem Decomposition of MIBO as UML class diagram with subsystems.

Blackboard Subsystem

The `Blackboard`, `Supervisor` and `KnowledgeSource` objects form the center pieces of the blackboard subsystem. Our implementation uses an object-oriented approach to query the blackboard for any `Hypothesis` that conforms to a particular interface that can

be processed by a `KnowledgeSource`. The blackboard provides a number of additional convenience methods (omitted in the diagram) to provide quick and convenient access to the stored information. Hash maps can be used to optimize the access for particular types and users. Despite some overall context information of the environment, most data on the blackboard is user-specific. We use a multi-tenancy approach for the blackboard which clearly separates user-specific data from each other in a kind of sandbox. This also reduces the number of relevant `Hypothesis` objects that have to be evaluated for each user and thus improves MIBO's performance.

Hypothesis

The blackboard keeps `Hypothesis` objects which contain information about an `Event` or contextual information of the environment. This approach has the advantage that the content information of the `Hypothesis` does not have to be subclassed or conform to a specific protocol but just have to be stored on the blackboard in a `Hypothesis`. A `Hypothesis` provides a number of attributes, such as a confidence score (how reliable is this hypothesis), a `createdAt` timestamp when the hypothesis has been created, a `validUntil` timestamp for real-time input data and it may also contain a list of references to other hypotheses that build up that hypothesis. Augusto states that timestamps are important to consider the provided input data either as complementary or independent [Augusto10b].

Timeout and Validity of Hypotheses

While some hypotheses about the user or the environment context may reside longer on the blackboard, the `Event` objects are typically valid for only a couple of seconds. If a speech or gesture event does not lead to an explicit command within a certain time frame, the `TimeKeeper` will remove the hypothesis from the blackboard. However, the validity of a hypothesis can be extended when it is incorporated in an ongoing interaction, e.g. if the user pointed at a light fixture and is currently about to dim that light using an up/-down gesture. The point-at hypothesis will remain for its default validity time frame after the interaction has stopped. That would allow a user to still refer to the same fixtures without repeating single commands.

Commands

The `CommandFactory` is used to create `Command` objects. Most controls can be realized by one type of command, the `SetItemValueCommand`. This command sets the value of an item to a specific value. Through the decomposition of fixtures into items, even more complex operations can be realized using this simple command. However, if necessary, additional subclasses of `Command` may be added easily. Similar to Wilson's approach [Wilson03], we suggest to decompose any command into the actual action ("set value to x") and the referent ("item y") of that action, to allow flexible combinations.

Command Execution

The `CommandExecutor` is a knowledge source that works on the highest abstraction level of the blackboard and searches for executable `Command` objects. These objects are then passed on to the `ItemController` to be physically executed in the smart building. Successfully executed commands are removed from the blackboard and put into the archive stack. This allows for the incorporation of context information, such as undoing of executed commands or taking actions that are based on previously executed commands. The `ItemController` provides an abstraction to be used with different brokers to control the physical fixtures in the space. Dependency injection is used to decouple MIBO from any specific instance of that brokers and enhance the interoperability. We use OpenHAB [2] as broker to provide abstractions to the numerous device protocols of the fixtures.

Composite of Fixtures

Composite of fixtures are required to group fixtures and address them as if they were only one. For instance, in a room with many light fixtures, the user might not want to turn on/off every individual light but rather a composition of several lights. A large room might consist of two to four light fixture composites, containing a total of 20 individual light fixtures. The same applies for heating/cooling units and window blinds. The composite pattern [Gamma95] approaches this issue by allowing applications to treat individual fixtures and compositions of these fixtures uniformly.

[2] http://www.openhab.org

Modality Agents

The `ModalityAgent` classes provide the link to the external devices and modality sensors. These modalities create `Event` objects of different types, e.g. for voice or gesture events. Events are usually composed of several device inputs (e.g. multiple fingers touching a display) that have to be interpreted from complex raw data, such as blob identification and analysis. This is the responsibility of the knowledge sources of each modality. It is up to the implementation of the modality to decide about how much sensor fusion and reasoning is performed on the actual sensor software or in the knowledge sources of MIBO. For example, a gesture modality could either start putting raw events on the blackboard, such as tracking data of a human skeleton which is then enhanced through the knowledge sources of the modality to recognize a raising hand and eventually a waiving gesture. Neural networks or hidden markov models are typically used to perform pattern recognition [Johnston97] within the knowledge sources. Based on our case studies, we suggest to perform event parsing and gesture recognition on the agent itself and put rather higher level data on the blackboard, such as recognized types of gestures. This simplifies the data flow, improves testability and optimizes performance. The particular input data from the distributed agents and sensors is received by each modality which provides for a simple stateless HTTP-based interface.

Standardized XML representations, like *EMMA* [Johnston09b] may be used for the communication between the distributed agents and the knowledge sources of each modality to enable the interoperability of HW/SW-components from different vendors and research sites.

4.3.3. Multimodal Fusion and Integration

The `MiboDefinitionExpert` is in charge of the integration of multimodal input events. It identifies the best potential interpretation, based on the interaction model. If a specified definition within the model is fulfilled completely, it issues a `Command` object which is then executed by the `CommandExpert`.

As described in section 2.6.2, there are three levels of multimodal fusion. All three levels can be performed using the MIBO framework. However, since our case studies involve rather loosely coupled modalities, we use decision-level fusion in our case studies. This still allows for mutual disambiguation of modalities, as described by Dumas [Dumas09]. For instance, as depicted in figure 4.7 on page 61, a user may point between a light and fan.

Even though the user may point with a slightly higher probability at the fan, the spoken sentence *"Dim that light"* can overrule the former decision and switch to the light as the referred fixture. The calculation of this fusion is based on N-best lists. Since every hypothesis is weighted with a confidence score, a point-at gesture would typically return a list of possible fixtures, along with confidence scores. The probability of each multimodal interpretation in the resulting set is determined by simple multiplication of the probabilities.

Temporal Integration

Multimodal input typically occurs with an offset in time. Empirical studies of multimodal interaction by Oviatt [Oviatt97] have shown that speech typically follows gesture within a time frame of three to four seconds while gesture following speech is rather uncommon. Temporal integration combines this data to a meaningful data set. López-Cózar and Callejas [López-Cózar10] differentiate three forms of temporal integration – microtemporal, macrotemporal and contextual fusion. Microtemporal fusion is carried out when the user inputs are complementary and overlap in time. Macrotemporal fusion is carried out when the inputs are complementary and belong to the same analysis window but do not overlap in time. Contextual fusion combines information without considering any time restrictions [López-Cózar10].

MIBO's software architecture, leveraging the blackboard pattern and timebound hypotheses allows for all three types of temporal integration. The `MiboDefinitionExpert` continuously evaluates the unexpired hypotheses and searches for a matching definition in the interaction model to execute a command. The `TimeKeeper` removes expired hypotheses from the blackboard if no matching definition for a control can be found within a particular time frame. Every hypothesis has its own defined validity time. This integration architecture ensures that multimodal input is integrated appropriately even when it is received at different points in time and in arbitrary order. Hypotheses with very long or infinite validity times would even allow for complex contextual fusion.

4.3.4. Multi-Device User Feedback

The functional requirement FR2 in section 3.4 formulates the need to provide feedback to users across devices. The requirement is based on our wizard-of-oz study (section 3.2) which shows the importance of providing feedback about the user's multimodal input. For example, when a user applies a point-at gesture to refer to a fixture, it is helpful to see

whether the system has recognized the correct pointing direction. In particular, our study shows that for fixtures that are not in close proximity (more than four meters distance), it becomes harder for users to hit the target correctly using a point-at gesture. This relates to the laws formulated by P. Fitts [Fitts54] about perceptual-motor functions of human-beings. Additionally, it is also important for a user to acknowledge whether a command (e.g. speech input) has been recognized correctly. Both mechanisms are used to actually meet the requirement of error prevention. The actual feedback may again be multimodal, involving acoustic, visual or haptic feedback. In fact, it may also be multi-device. In our case studies in chapter 6, we show examples of providing feedback through MIBO's architecture on a smartwatch, smartphone and by mounted room speakers.

MIBO provides feedback by leveraging the observer design pattern to realize a subscriber/publisher paradigm. This pattern decouples the yet unknown subscriber from the actual framework and reduces latency for feedback distribution. MIBO publishes different kinds of events that authorized devices can subscribe to. For example, in our NICE control case study in section 6.2, we demonstrate a smartwatch subscribing to changing point-at events. This allows users to point at arbitrary fixtures in the room and to see a picture and label of that fixture on their wrist. MIBO allows subscribing to the events of an individual user on a separate channel or to entire namespaces at once.

To realize a scalable feedback architecture, MIBO leverages the capabilities of WebSockets [3]. Once a regular HTTP request has been set up, WebSockets can upgrade that request to an open socket connection. This removes the overhead of HTTP requests, such as headers and meta information, and thereby reduces network load and enhances performance significantly.

4.3.5. Access Control and Security

Access control in smart buildings is often based on trade-offs. On the one hand, an office worker should not be restricted to turn off a light in a colleague's office space, if it has been forgotten. On the other hand, potentially sensitive context information must be protected and must not be retrieved by ineligible persons. For guests, there must be a simple zero-configuration approach where guests may use functions that they are granted access to. In general, any information which enters or leaves MIBO, is bound to a user that it belongs to. Authentication mechanisms like a shared secret or certificates can be used to prove the authenticity of each client that interacts with MIBO. While this works well on personalized

[3] IETF Standard (2011), RFC 6455

smartphone or smartwatch devices, it may be a problem when sensors like cameras are mounted in the room. If these devices do not have the capability to identify a user, they may use a unique generated username per namespace for each person that interacts with it. In most cases, this is acceptable because the user must have access to the room, so the access control is comparable to a traditional light switch. In the future, face recognition may be used to overcome this issue. We propose a strategy, where the criticality of an action determines the explicitness of an authentication.

To allow for any access control, the blackboard as well as the published feedback is based on the associated username of the input. Despite some overall context information of the environment, most data on the blackboard is user-specific. This user-specific data is separated completely from each other.

4.3.6. Design Rationale

In the system design of multimodal controls for smart buildings, the developer has to identify the subsystems to be mapped to the user client and the ones to be mapped to a central broker system, dedicated to the whole space. As depicted in figure 4.10, Schiele et al. differentiate between "smart peers" and "smart environment" [Schiele10]. We consider the smart peers to refer to a fat client and the smart environment to a fat server (and a number of thin clients).

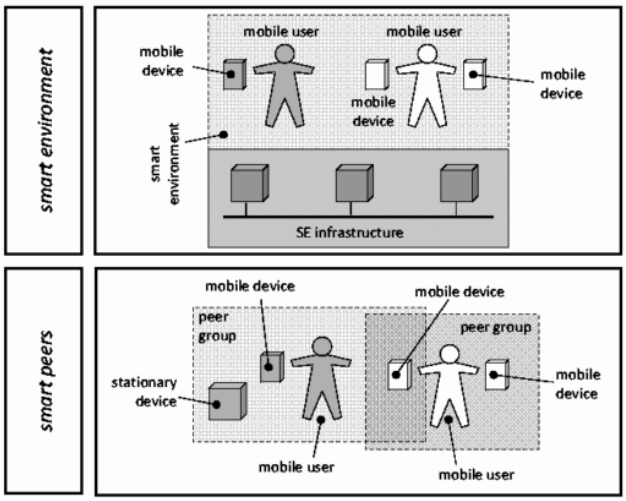


Figure 4.10.: Smart peers and smart environments as organizational models. Source: Augusto, [Augusto10b].

MIBO fosters an approach with a fat server and numerous thin clients. This allows for the centralized configuration of the interaction model and enables adaptions without touching the clients. In this case, the clients serve as sensors for multimodal input. This input is preprocessed to an appropriate level (depending on the modality and use case) and then forwarded to MIBO. It is yet unknown to the client, whether this input yielded to an actual control process of a fixture because this may depend solely on the interaction model and input from other modalities and devices. However, the client may subscribe to a feedback channel, to provide feedback to users. This approach allows for the seamless integration of input and output data across multiple modalities and devices. For example, if a gesture or speech recognizer should only be activated on the button press of a watch, this could be added easily to the particular MIBO definition in the interaction model. In the case of a fat client, this would require the actual gesture/voice sensor application to be changed. Additionally, the approach of a thin client provides a number of advantages for abstraction. It enables the abstraction of the numerous protocols that are being used in smart buildings, as well as the abstraction of context data, such as weather, traffic or personal calendar.

In MIBO, the interaction model consists of MIBO definitions that are represented using *MiboML*, which is described in the following chapter.

5. MiboML – A DSL for Multimodal Interaction Models in Smart Buildings

Customization and extensibility are major requirements for multimodal intuitive building controls, as identified in section 3.4. To allow for the definition of interaction models at runtime, we have developed MiboML, which is a domain-specific language (DSL). MiboML is used to model the multimodal interaction of human-beings with fixtures in a smart building and declaratively define the multimodal integration strategy. Johnston showed in general that the structure of multimodal commands can be represented declaratively in a multimodal context-free grammar, being interpreted by a weighted finite state automaton [Johnston00] [Johnston05]. However, Johnston did not link this representation with semantic commands that are executed at a particular user input. This is an integral part of MiboML, eventually allowing end-users to configure multimodal interactions in buildings at runtime. The Meta Object Facility (MOF) [OMG06], introduced by the Object Management Group (OMG), consists of four layers to describe meta models and their instantiations. We consider MiboML to be a MOF-compliant meta-model on layer M2, which allows the instantiation of interaction models for buildings (see figure 5.1).

- **Layer M0** represents an interaction performed by a user in the real world, e.g. a user pointing at a fixture. The interaction is provided by a user.
- **Layer M1** represents the interaction model for a user in a particular building. It can be represented by a grammar derivation string of MiboML. It is provided by a developer or facility manager.
- **Layer M2** represents MiboML, the meta-model for multimodal interaction models in buildings. It is provided as grammar by the MIBO framework.
- **Layer M3** represents the meta-meta model and is provided by the Meta Object Facility.

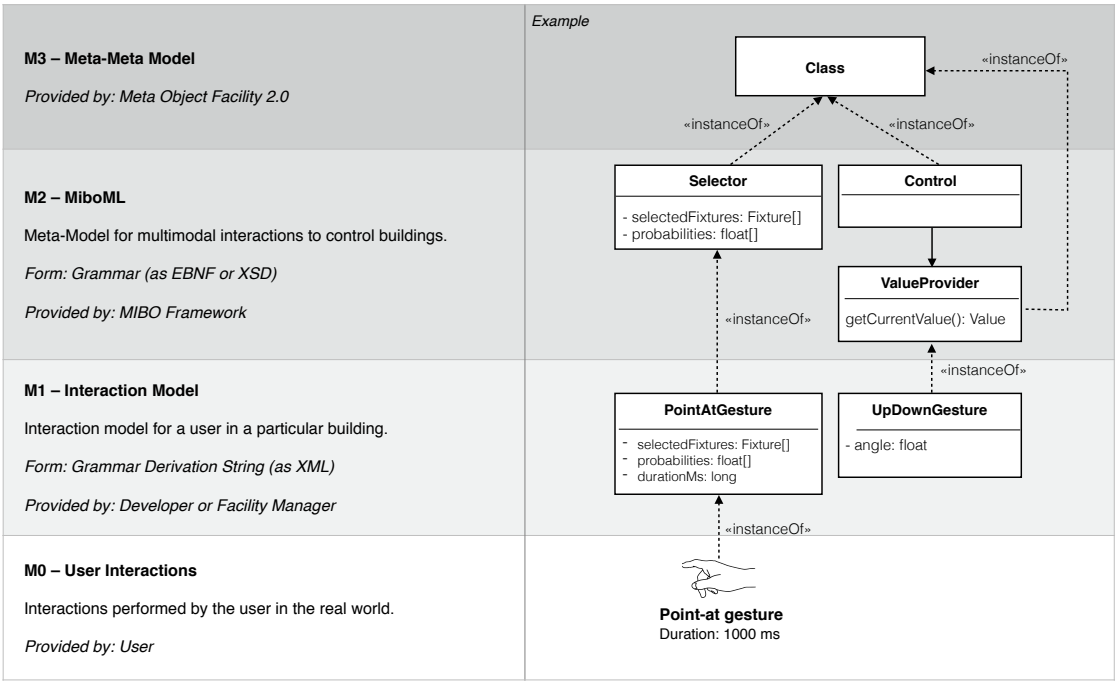


Figure 5.1.: MiboML at layer M2 in the Meta Object Facility.

5.1. Design Goals of MiboML

Based on the requirements identified in chapter 3, we derived design goals for MiboML. Following our formative research approach, the case studies described in chapter 6 have iteratively influenced these design goals.

Adaptability. Due to flexible project-related assignments of teams to rooms in commercial buildings and frequent rearrangement of equipment in buildings, controls need to be adaptable without additional software development efforts. As described in requirement FR3 in section 3.4, we distinguish three levels of customization for interactions in buildings – building specific, cultural and individual customization. In order to realize this requirement, we propose an object-oriented approach using the concept of inheritance and overriding of interactions. Per default, every occupant *inherits* the predefined definitions for interactions in a building. An occupant may then *override* particular definitions and replace them with individual ones. For example, if a motor impairment prevents users from performing a particular gesture, they may override this gesture with either another

gesture or even an interaction from another modality, such as voice or gaze.

Flexible multimodality. Our user studies described in chapter 3 have shown that occupants prefer different commands to control their environment, depending on the actual context. Therefore, MiboML has to allow for flexibility. For example, all following inputs by the user may result in a light turning on:

- Say *Turn on the light*.
- Say *light* and perform an *up* gesture.
- Point at the light and say *turn on*.
- Point at the light and perform an *up* gesture.
- Say *Turn on these lights* and then point at one or several lights.

Even within one modality, there must be alternatives. While some people might prefer the term *turn on*, some may actually prefer *switch on* or entirely different expressions. Wilson and Shafer have pointed out similar requirements to combine modalities flexibly [Wilson03].

Modality Independence and Extensibility. As research in input and output modalities continues to evolve, MiboML must be extensible to deal with new modalities. For example, smart glasses may provide reliable gaze-tracking features in the future to enable the selection of fixtures through the eyes. Research in brain-machine interfaces [Lebedev06] [Wolpaw02] may allow to execute commands without gestures or speech by just activating particular regions of the brain. The inclusion of these modalities must be possible by simply appending additional vocabularies that can be maintained separately. Adding modalities must not require changing the core grammar of the language.

Readability. A language for the description of human-computer interaction such as MiboML should be readable for both, human-beings and computers. While humans have to use the language to define their preferred interactions, computers have to be able to process the interaction model efficiently in polynomial time.

5.2. MiboML Grammar

In section 4.2, we have identified the parts of an interaction to control fixtures, consisting of a scope, entry condition, select, control and exit condition. These definitions can be expressed as a context-free grammar *G* in *Extended Backus-Naur Form (EBNF)*[1].

5.2.1. Core Grammar

The following section describes the core grammar of MiboML which enables the instantiation of interaction models for buildings. It is shown in listing 5.1 below and follows the model of figure 4.6, described on pages 57ff.

Listing 5.1: MiboML Core Grammar G in EBNF.

```
G = (T, N, P, Definitions)} with the following productions P:

Letter :== "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" |
    "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" |
    "X" | "Y" | "Z"
Digit :== "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
Identifier :== Letter+
Number :== Digit+ ["." Digit+]

Definitions ::= Definition+
Definition ::= Scope Entry Select Control Exit

Scope ::= (Group | Item)+
Group ::= "ItemGroup" Identifier
Item ::= "Item" Identifier

Entry ::= Trigger*
Exit ::= Trigger*

Select ::= (Selector ["optional"])+

Control ::= (Set | Toggle | Increase | Decrease)+
Set ::= SetTo | SetContinuous
SetTo ::= "set to" Number "on" Trigger
SetContinuous ::= "set" ValueProvider
Toggle ::= Trigger+
```

[1] Syntactic metalanguage - Extended Backus-Naur Form (EBNF), International Standard ISO/IEC 14977

```
Increase ::= (IncreaseByAbsolute | IncreaseByValueProvider)
IncreaseByAbsolute ::= "increase by" Number "on" Trigger
IncreaseByValueProvider ::= "increase" ValueProvider
Decrease ::= (DecreaseByAbsolute | DecreaseByValueProvider)
DecreaseByAbsolute ::= "decrease by" Number "on" Trigger
DecreaseByValueProvider ::= "decrease" ValueProvider

Selector ::= GestureSelector | VoiceSelector | GazeSelector
Trigger :== GestureTrigger | VoiceTrigger | GazeTrigger
ValueProvider ::= GestureValueProvider | VoiceValueProvider
```

The modalities (here: Gesture, Voice and Gaze) may then provide their own productions, depending on their available actions and required attributes. Their dedicated start symbols just have to be added to the productions ValueProvider, Selector and Trigger.

5.2.2. Modality Extension Grammar

The following productions in listing 5.2 illustrate the extension concept of MiboML by examples. Each modality defines its own events and the associated attributes.

Listing 5.2: MiboML Grammar (Modality Extension) in EBNF.

```
GestureSelector ::= "gesture:PointAt" [Duration]
Duration ::= Number
GestureTrigger ::= "gesture:ReccuringHand" | "gesture:LeftHandOverHead"
GestureValueProvider ::= "gesture:UpDown"

VoiceSelector ::= "voice:referToFixture" | "voice:referToFixtureType"
VoiceTrigger ::= "voice:turnOn" | "voice:turnOff" | "voice:dim"
               | "voice:confirm" | "voice:open" | "voice:close"
VoiceValueProvider ::= "voice:setTo"

GazeSelector ::= "gaze:lookAt"
GazeTrigger ::= "gaze:eyeBlink"
```

5.2.3. Grammar Parsing

The MiboML grammar allows the formulation of interactions in a building. For example, the following parse tree in figure 5.2 shows a definition to dim lights by pointing at the light for at least one second and using an up/down gesture to set the light level. Voice

commands have to be used to start and stop the dimming process. This example has been visualized earlier in figure 4.7 on page 61.

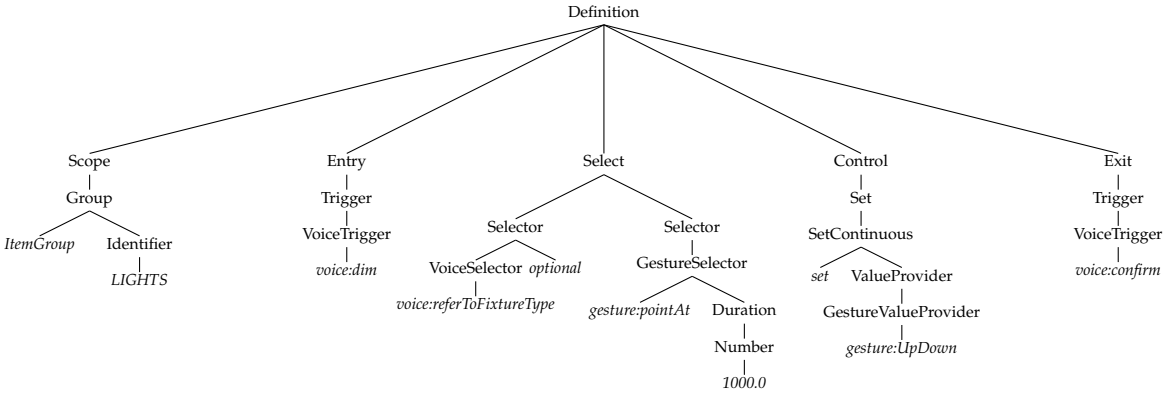


Figure 5.2.: Parse tree for a definition to dim lights using gestures and speech.

MiboML is a context-free language which can be parsed by a deterministic pushdown automaton. For each definition of the interaction model expressed in MiboML, finite-state machines can be generated to interpret the multimodal user input. Appropriate actions are then taken in the control state (CRL), based on the semantics given in section 4.2. Figure 5.3 shows a generated finite-state machine for the above interaction definition to dim a light using the combination of gestures and speech.

5.3. Lexical Representation

For the reasons of extensibility, readability and maintainability, we use the Extensible Markup Language (XML) [Bray06] as lexical representation for MiboML. This design decision was based on the goals described in section 5.1. XML provides several advantages in the context of MIBO.

Extensibility. XML offers the concept of namespaces, which allows developers of new modalities, to extend the language separately from its core grammar. For example, new gesture or voice events may be added easily. Additional attributes may be added as needed, such as *minimum duration*, *maximum fixture distance*, *maximum pointing deviation* for a point-at gesture. Existing languages to describe interactions in these modalities may even be integrated smoothly in their own namespaces, such as VoiceXML [McGlashan04],

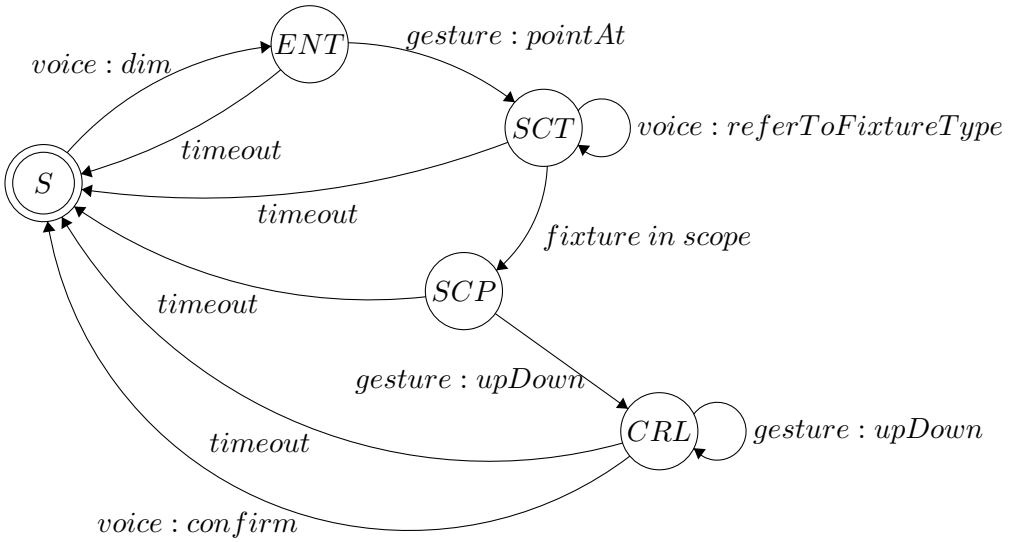


Figure 5.3.: Generated finite-state machine to interpret multimodal user input to dim a light.

GestureML[2], EmotionML [Burkhardt14], InkML [Watt11] and others. For example, GestureML would allow developers to define expected gestures in a detailed manner.

Unambiguousness. In general, context-free grammars bear the risk to be ambiguous. XML solves this issue by providing a concept for tag separation and escape characters.

Readability. Our goal is to provide developers and expert users a readable representation of interaction models in MiboML. At the same time, it must be easy to process for machines. Both, the readability for humans as well as for machines are among the top ten design goals of XML:

- "XML documents should be human-legible and reasonably clear." [Bray06]
- "It shall be easy to write programs which process XML documents." [Bray06]

XML also provides convenient ways to embed comments.

[2]http://www.gestureml.org/

Finally, the discussed definition from figure 5.2 to dim a light using voice and gestures, is represented in MiboML, as follows in listing 5.3.

Listing 5.3: Definition to dim a light in XML representation

```
<definition id="1" name="dimLights" active="true">
    <scope>
        <item group="Lights" />
    </scope>

    <entry>
        <voice:dim />
    </entry>

    <select>
        <gesture:pointAt minDuration="1000" />
        <voice:referToFixtureType optional />
    </select>

    <control>
        <set>
            <gesture:upDown />
        </set>
    </control>

    <exit>
        <voice:confirm />
    </exit>
</definition>
```

5.4. Conflict Resolution

Interaction models can be inconsistent when they contain conflicting definitions. For example, when a gesture command exists in a unimodal and multimodal definition, the unimodal definition would be executed immediately, even if additional speech is still being processed at the same time. Johnston approaches this issue by introducing a delay: "If speech does not follow within the three to four second window, or following speech does not integrate with the gesture, then the unimodal interpretation is chosen. This approach embodies a preference for multimodal interpretations over unimodal ones, motivated by

the possibility of unintended complete unimodal interpretations of gestures." [Johnston97] However, Johnston used multimodal commands in map-based navigation, which is a different application domain. Our wizard-of-oz study, described in section 3.2, showed that today's occupants of smart buildings in average accept a maximum delay of one to two seconds until they try to repeat a command. Therefore, Johnston's suggested delay is not acceptable in smart buildings. Instead, we suggest keeping definitions consistent, such that two definitions do not cause a control command at the same conditions. This is defined as follows.

Let $Trigger$ be the set of all defined MIBO triggers (e.g. "turn on" speech command).
Let $ValueProvider$ be the set of all MIBO value providers (e.g. up/down gesture).
Let $Selector$ be the set of all defined MIBO selectors (e.g. point-at gesture).
Let $Users$ be the set of all defined users.
Let $AllDefs$ be the set of all declared MIBO definitions
Let $defs\colon Users \longrightarrow 2^{AllDefs},\ u \longmapsto UserDefs$
where $UserDefs$ is the set of definitions for user u

Let $2^{Trigger}$ and $2^{Selector}$ be the power sets of $Trigger$ and $Selector$
Let $e \in 2^{Trigger}$ be an arbitrary entry set
Let $s \in 2^{Selector}$ be an arbitrary select set
Let $c \in 2^{Trigger \cup ValueProvider}$ be an arbitrary control set
Let $Control(d, e, s, c)$ be the predicate to determine if
e, s and c cause a control output for definition $d \in AllDefs$.

For a given $u \in Users$, two definitions $d, d' \in UserDefs$ are inconsistent, if and only if:

$$\exists e \in 2^{Trigger}, \exists s \in 2^{Selector}, \exists c \in 2^{Trigger \cup ValueProvider} : Control(d, e, s, c) \wedge Control(d', e, s, c)$$

Thus, we consider two MIBO definitions to be inconsistent when there exist sets of entry, select and control conditions that cause two definitions to produce a control output at the same time. Even though, there may be theoretically, reasonable scenarios where two definitions increase/decrease the value of a fixture at the same time, we do not recommend or support these types of definitions because of maintainability. The following measures are taken to reduce inconsistent MIBO definitions.

Prioritization of definitions. For two definitions $d, d' \in Userdefs$ with the defined entry sets $e \in 2^{Trigger}$ and $e' \in 2^{Trigger}$, where $e \subset e'$ and $Control(d, e, s, c) \wedge Control(d', e, s, c)$, definition d' is considered to be the more complex one and prioritized over d. That means, for entry set $e' \in 2^{Trigger}$, d' is executed and d is ignored.

Optional interactions. Many situations that would cause inconsistent definitions can be resolved by introducing *optional interactions* within one definition. MIBO allows marking selectors to be optional. This approach provides better maintainability without compromising the expressive power of MiboML. For example, a definition might include a mandatory point-at gesture and an optional `voice:referToFixtureType` selector, as seen in listing 5.3 on page 80. That means, users might further specify the fixture type that they point at (e.g. by adding the spoken word "Light"). This helps MIBO to determine the focused object if the point-at gesture would result in two fixtures of different types next to each other. However, if a user prefers to omit this hint or factors like environmental noise absorb that speech command, the definition is still executed with the remaining information.

Customization. Customization not only allows the extension of existing interactions but also actively overrides or disables existing definitions. For example, a patient with motor impairment might want to exclusively use a unimodal speech command instead of a multimodal command that includes gestures. The unimodal definition might be inconsistent with the default multimodal command and is therefore available exclusively for the patient.

Interactive debugging. To provide the authors of MIBO definitions with useful debugging opportunities, we introduce a meta user interface for the dynamic definition of multimodal controls, described in more detail in the following section 5.5. It generates warnings and errors, whenever a user is about to set up an inconsistent definition by checking if the new definition is inconsistent with any other definition of that user.

5.5. IDE for Multimodal Interaction Models

MIBO allows for the creation and adaption of multimodal interaction models at runtime. MiboML provides a meta model and a domain-specific language to describe these inter-

action models using an XML syntax. However, MiboML targets developers and is therefore not intended for end-user customization. Though, this customization is important to adapt the interaction model to the context of the user's culture, building (room layout, available fixtures, available sensors) and to their own capabilities due to physical limitations. Furthermore, debugging, simulation and conflict-resolution should be supported by an integrated development environment (IDE) for MiboML. To meet this requirement, we developed an IDE that allows end-users to create and adapt interaction models based on MiboML visually using drag & drop without programming skills. Figure 5.4 shows the IDE with the debbuging console on the top, and a visual programming tool for interaction models below. The toolbar on the right side allows dragging elements of the model from a library of fixtures and modalities. Finally, MiboML code is compiled from the defined interaction model.

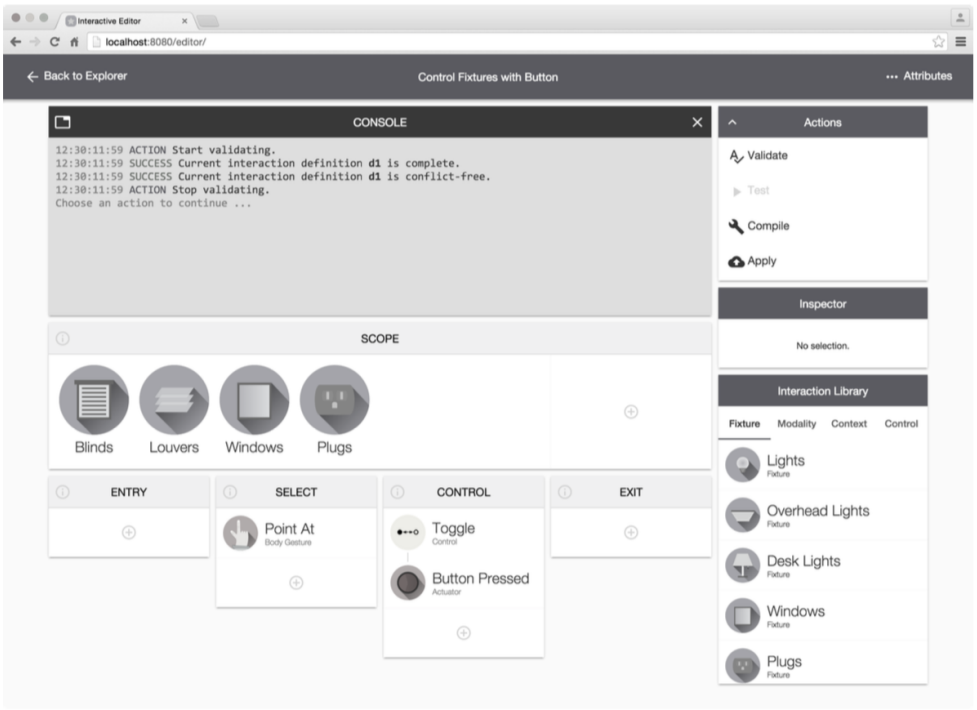


Figure 5.4.: The IDE for MIBO interaction models. Source: [Johanssen15].

To provide platform independence, the main component of the IDE is implemented as a web application. Furthermore, by utilizing an additional server-side component, it is extensible and dynamically adapts to updated MIBO meta models at runtime. The IDE

supports the validation and debugging of interaction models as well as a heuristic to determine inconsistent interaction definitions that are either identical or obfuscate each other.

A user study by Johanssen and Peters [Johanssen15] revealed that participants were able to create suitable interaction models from a pool of more than half a million possible combinations after an introduction of less than ten minutes. Moreover, each participant showed a significant improvement in task completion time when using the Interactive Editor, comparing the initial contact and the last performed action within the Interactive Editor.

6. Case Studies

To test the feasibility of MIBO and to evaluate the performance of multimodal intuitive building controls, we conducted three case studies for multimodal controls. These case studies have been developed and continuously tested in the Robert L. Preger Intelligent Workplace [Hartkopf97] at Carnegie Mellon University in a formative approach. In addition, the generalizability of the results to other buildings and environments has been demonstrated in the smart lab and iTuepferl at Technische Universität München, and in a corporate environment, the Capgemini Innovation Lab.

The case studies incorporate *HomeGestures* – a gesture-based smartphone control, *NICE* – a hands-free natural user interface, and *SISSI* – a smart interface for speech service integration.

6.1. HomeGestures – A Gesture-based Smartphone Control for Smart Buildings

In [Peters11], we presented HomeGestures – a gesture-based approach for controlling various fixtures in instrumented buildings using a smartphone. The user simply points the smartphone at target objects and completes specific gestures. For example, pointing at the top of a window and making a down-gesture is interpreted as a command to lower the blinds at that window. Pointing the device at a light fixture and making an up-gesture raises the light levels. The implementation of the point-and-gesture control is based on the magnetometer, gyroscope and accelerometer built into recent smartphones. In combination with addressable fixtures and wireless infrastructures, this controller reveals how a wide variety of fixtures in a building can be controlled intuitively by pointing and gesturing.

6.1.1. Interaction Design

The visionary scenario of a gesture-based control interface in a smart home has guided this case study. However, we wanted to gain stronger indications of the extend to which gestures can be used intuitively for controlling smart home fixtures. Therefore, a usability study which is loosely based on Wobbrock's guessability approach (see section 3.1) was conducted. The study consisted of ten user experiments. During each experiment, an individual was asked to perform specific tasks such as turning on and off a light or opening window blinds using a wand (see fig. 6.1).

Figure 6.1.: The experiment showed anecdotal evidence that people used pointing and gesturing.

Ten test persons working in a smart office environment at Carnegie Mellon University were part of the study. To avoid bias, the subjects did not know about the objectives of the study. Each subject was given a wand and was asked to open two different window blinds and turn off a light at the ceiling using the wand. The blinds were faked using movable walls and labeled with the word "Blinds". The light was depicted using a painting of a light at the ceiling of the room. It was observed that nine out of ten persons pointed at the fixtures and combined that with an *up*-gesture to raise the blinds at the window and a *down*-gesture to turn off the light. One person included speech for the commands. In the experiment, we have selected the wand in particular to indicate by the means of anecdotal evidence:

- Modality: Would occupants intuitively use gestures to control fixtures in a room?
- Select gesture: Do occupants use the pointing gesture to select objects for controlling?

- Control gesture: Which gestures are appropriate for performing up/down and on/off commands?
- Speech command: Do occupants combine their wand gestures with a *spell*? If yes, what is the speech command like?
- Eye contact: Do the occupants look at the objects that they control?

The guessability experiment showed that people used pointing and gesturing to control fixtures in room. To verify whether the gesture-based control mechanism could be transferred from the wand to a mobile device, the experiment was repeated with an iPhone made of paper. The result showed that eight people continued to use these gestures, one was adding voice commands to the gestures and only a single test person was trying to use buttons on the fake iPhone's display. All persons looked at the target object during their gesture. Finally, the test provided sufficient indication to build an iPhone app prototype for controlling fixtures using gestures.

6.1.2. Implementation

The smartphone controller leverages the built-in magnetometer, accelerometer and gyroscope for identifying the orientation of the device and for capturing gestures such as an up- and down movement of the user's arm. The resulting gesture control for smart environments is presented by Peters et al. [Peters11].

Identification of the Smartphone's Orientation

Since occupants can use their smartphone to *point* at fixtures, the orientation of the device has to be determined. This is approached by performing a sensor fusion of the smartphone's built-in accelerometer, gyroscope and magnetometer. Every fixture is associated with a characteristic set of euler angles (yaw, pitch and roll), depending on the device's current orientation. In fact, the roll value is neglected because it does not influence the pointing direction. The initial euler angles of a fixture are recorded by pointing the smartphone at the particular fixture from a characteristic position of the room (e.g. the middle of the room or the position of the desk). The following sensors of the smartphone are used to determine the euler angles:

1. **Accelerometer.** The accelerometer is an electromechanical sensor which measures proper acceleration forces. These forces can be static, like the constant force of gravity, or dynamic – caused by moving the accelerometer. The angle that the device is tilted with respect to the earth can be measured by the amount of static acceleration through gravity. However, measurements are often blurred by a mixture of gravity, free fall and linear acceleration. Additionally, the accelerometer alone is subject to jitter. These circumstances harden the identification of the exact movement.

2. **Gyroscope.** The gyroscope is a sensor for measuring orientation, based on the principles of conservation of angular momentum. It leverages the coriolis effect which is a deflection of moving objects when they are viewed in a rotating reference frame. Compared to the accelerometer, the gyroscope is accurate on short term but drifts on long term. The drift can be eliminated when the gyroscope data is fused with the accelerometer using a Kalman filter. The gyroscope is optional in the implementation of HomeGestures but improves the agility of the control because of the short term accuracy.

3. **Magnetometer.** The magnetometer is a sensor to measure the strength or direction of a magnetic field, such as the Earth's magnetic field. It is the most important sensor in the HomeGestures implementation because it can be used as a compass to provide a stable reference frame for the orientation (the yaw value). Unfortunately, the magnetometer is influenced by external magnetic sources and even the phone itself. The sensor fusion with the accelerometer and gyroscope can smooth the results and reduce jitter.

The magnetometer provides compass values for the orientation of the smartphone. However, the influence of external magnetic sources is problematic because the recalibration (moving the device in a figure-8 motion) disrupts the user's actual workflow. We suggest an approach to improve the usability by merging the magnetometer with the gyroscope in an automatic calibration. Figure 6.2 shows the developed automatic calibration of the system and the transitions between the states as UML state chart.

When the application becomes active, it begins in an uncalibrated state and waits for ten values given by the magnetometer with a minimum accuracy of 40. The accuracy property represents the potential error between the reported value and the actual direction of magnetic north. Thus, the lower the value of this property is, the more accurate is the heading. The system has to be blocked in this state because the orientation of the phone is

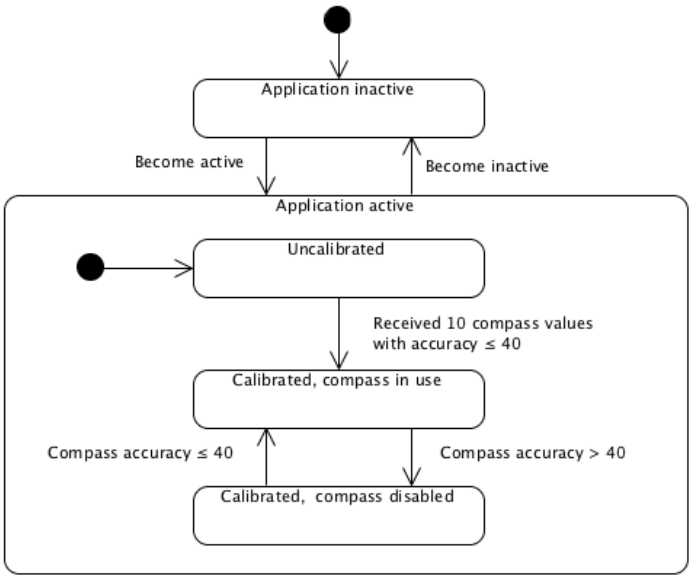


Figure 6.2.: Calibration states as UML state chart.

still unknown at that time. After the compass provided an initial orientation, the gyroscope is able to take over whenever the compass accuracy becomes higher than 40. This behavior supports the usability of the system because once the system is calibrated, the usage is no longer disrupted, even if the compass is influenced by external magnetic sources.

Sensor Fusion

The gyroscope is fused with the accelerometer to eliminate its drift. This first fusion step can be performed using a Kalman filtering and is often provided by the hardware manufacturers. This value is called "device motion". The yaw angle of device motion needs to be fused with the magnetometer yaw to establish a stable reference frame because the accelerometer and the gyroscope can only capture yaw deltas and not the absolute value. The magnetometer provides the long-term reference frame, while the fused accelerometer/gyroscope captures agile high-frequency moves. For this scenario, Paul C. Glasser suggests a low-pass/high-pass complimentary filter (see figure 6.3).

The short-term yaw values of the accelerometer/gyroscope are immediately added to the fused value. On the long term, the fused value always converges towards the yaw value of the magnetometer. We also implemented various approaches to combine this with indoor positioning, e.g. using WiFi fingerprinting, iBeacons and ultrasound positioning.

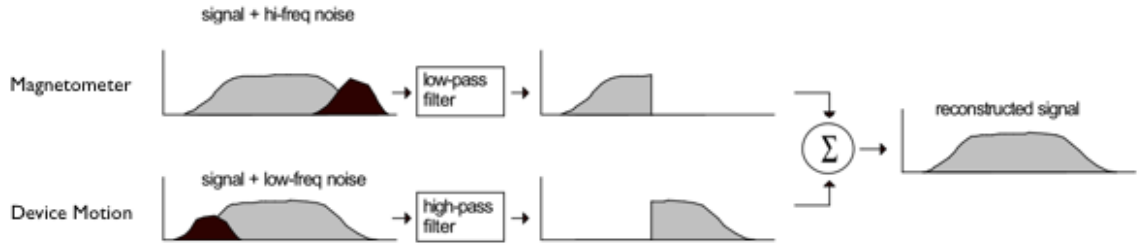


Figure 6.3.: Based on: Glasser, "An Introduction to the Use of Complementary Filters for Fusion of Sensor Data" [Glasser11].

Since the exact position within the room cannot be identified, there will be deviations in the measurements when the user moves to different locations in the room. However, tests have shown that this principle is reasonable for rather small rooms with equally distributed fixtures and a maximum number of about seven fixtures per room.

Migration to MIBO

After HomeGestures' initial implementation [Peters11], it has then been upgraded to leverage the MIBO framework. Thereby, a major part of the application logic has been shifted away from the smartphone controller. While the initial version contained hard linking between a recognized gesture and a fixture-specific command, this complexity was later reduced to the aspect of recognizing a gesture and sending that result to MIBO. For instance, the logic that an up-gesture should raise a blind when the user points at a window is now reflected by MIBO and configured in an interaction model using the domain-specific language MiboML.

6.1.3. Related Work

Wu et al. use an Android phone that acts as a physical ubiquitous interaction device in the real world, called MagicPhone [Wu10]. It senses the pointing orientation using a built-in accelerometer and magnetometer. It uses both, gestures on the device's display and accelerometer-based hand gestures. In HomeGestures, the idea is enhanced in several ways. First, the sensors are fused with additional gyroscope data using a combination filter. This increases the responsiveness of the inert magnetometer. Second, the sensor data is combined with WiFi positioning for controlling multiple rooms without any additional user inputs. Our case study also investigated approaches for error prevention, error recov-

ery and the difficulty of target objects covering each other (objects behind other objects). Finally, MIBO allows the combination and integration of arbitrary modalities and decouples the interaction model from the implementation.

Tsukada and Yasumura [Tsukada04] introduced an interface for mobile environments, called *Ubi-Finger*. The interface enables users to control various fixtures in the real world using finger gestures. The *Ubi-Finger* itself is an input device worn on the fingers. It is connected to a PDA or laptop by a wired serial connection. The target fixture to be controlled is defined by infrared sensors attached to each fixture. The actual control commands are transmitted via W-LAN and executed by a server in the background. The gesture recognition is started and stopped using a small button (touch sensor). A target device is then selected by pointing. Afterwards, the device can be controlled with micro-gestures of fingers, like *pushing a switch*, *turning a volume knob*, and so on. Tsukada and Yasumura's ideas of using pointing and gesturing control mechanisms support the notion of gesture-based control interfaces. However, their approach requires the installation of additional infrastructure (infrared sensors) on each fixture. Apart from the visual detraction, this solution implies additional costs for the electronic pieces and labor for its installation. Second, the prototype finger is still too big and inconvenient for daily work. The fact that it has to be wired to a PDA restricts its mobility additionally. [Tsukada04]

Yang et al. [Yang07] describe a human-friendly HCI system to control home appliances. The proposed system makes use of two cameras enabling users to select home appliances by a pointing gesture. The system finds relative positions between two cameras through a visual pattern being attached on the side of one camera. With this information, the three-dimensional positions of objects are computed. The pointing command is defined as stretching the user's hand towards an object such that the center of the object is concealed by the hand. Since the pointing vector does not always hit an object exactly, the closest identified object is considered instead. The discussed use case is tailored to old and disabled people and requires the installation of cameras. A large amount of cameras would be necessary to cover every niche of a large building. Indeed, users might feel uncomfortable with this "big brother"-scenario. A similar approach, based on cameras and recognizing hand gestures, was later proposed and realized by Toth and Varkonyi-Koczy in 2009 [Tóth09]. [Yang07]

Adler and Reynolds [Adler07] also consider pointing as a natural method to indicate a focus of attention. They describe the use of mobile phones with cameras as pointing devices to interact with smart environments. The user selects objects by taking pictures of them. The recognizable objects contain visual image-processing hints, such as fiducial marks. Additionally, the authors describe the design of a location-aware service discovery protocol and the integration of the protocol with image capture and analysis. Though, their visual-based solution shows some weaknesses. Different light settings are problematic, e.g., if a light should be turned on in a dark room. Additionally, in a flexible space, objects may be covered by others (e.g. a flower in front of a heating) or objects may be so close to each other, that the camera captures more then one fixture in its picture. Also, the image-processing hints such as fiducial marks have to be attached to each addressable fixture. That requires labor and might constitute a visual distraction. [Adler07]

The *GeoWand* [Simon07] uses GPS data and a magnetometer/accelerometer to provide information on outdoor points of interest. Using a 3D block model of the environment, the system computes nearby geographic features (points of interest) that are visible from the user's current location. For the proposed application, it has to be considered that GPS is not available in indoor environments. Also, static 3D models can be problematic in flexible buildings, as fixtures such as floor lamps and even whole desks can be moved and changed frequently. [Simon07]

Kela et al. [Kela05] study accelerometer-based gesture control as a supplementary or an alternative interaction modality [Kela05]. It is not about pointing at fixtures but recognizing gestures (e.g. drawing a letter in the air) and linking them to a specific command. Two user studies are discussed. The first study concerns finding the right gestures for controlling a TV, VCR and lighting. The second user study evaluates the value of the gesture modality compared to other interaction modalities. The results of Kela et al. were used to identify the most intuitive gestures during the interaction design of HomeGestures. [Kela05]

In 2009, Bhuiyan and Picking [Bhuiyan09] published a review of the history of gesture controlled user interfaces to identify trends in technology, application and usability. Their findings conclude that gesture-based user interfaces afford "realistic opportunities for specific application areas". Bhuiyan and Picking consider "rich user interface using gestures as

appropriate for current and future ubiquitous and ambient devices". They summarized the history starting from early handwriting gestures on the stylus device in 1986 up to current commercial products such as the Nintendo Wii and up-to-date research. The summary is enclosed in Appendix B on page 133ff. [Bhuiyan09]

6.1.4. Evaluation

	1 - Strongly Disagree	2 - Disagree	3 - Neutral	4 - Agree	5 - Strongly Agree	Total	Avg.
It is fun to control fixtures using HomeGestures.	0	0	1	6	10	17	4.5
HomeGestures helps me to retain control of the fixtures in my office.	0	0	3	4	10	17	4.4
I am feeling more comfortable in my office with HomeGestures.	0	0	3	6	8	17	4.3
HomeGestures helps me reducing energy.	0	1	5	6	5	17	3.9

Table 6.1.: Questions and results on the performance of HomeGestures.

We evaluated HomeGestures in an observational study throughout four years. Over these years, approximately 20 users have decided on their own to use HomeGestures to set their preferred light and shading levels in the Intelligent Workplace [Hartkopf97] at Carnegie Mellon University. In an anonymous online survey, we could reach out to 17 of these users to assess the performance of the controller. Ten of them have been using HomeGestures since the first release in the year 2011. Twelve of all questioned users have used HomeGestures at least weekly, seven of them even stated daily usage. In the survey, we asked users about perceived advantages of the controller using a 5-point Likert scale [Likert32]. The questions and answers are shown in table 6.1.

The survey shows that 94% of our users is having fun while controlling fixtures using HomeGestures. Also, the majority of users agrees or strongly agrees that HomeGestures helps to retain control of the fixtures in their office and that users feel more comfortable in their office with HomeGestures. None of the participants disagreed on either of these three categories. With respect to energy savings, 64.7% of the users agree or strongly agree that HomeGestures helps them reducing energy. One participant disagrees on this matter while the rest stays neutral.

As the user base of HomeGestures was growing over time, we observed that users learn to use these kind of controls by observing another person who is performing the gestures.

We wanted to get more indication on this aspect using the questions shown in table 6.2.

	1 - Strongly Disagree	2 - Disagree	3 - Neutral	4 - Agree	5 - Strongly Agree	Total	Avg.
I mainly learned how to use HomeGestures by observing how another person gestured with it.	0	3	1	7	5	16	4.6
Trying out the gestures by myself was very important for me in the learning process.	0	1	3	6	6	16	4.1
When I tried out HomeGestures for the first time, I basically knew how to interact with it.	0	0	4	6	6	16	4.1

Table 6.2.: Questions and results on the learning aspect of HomeGestures.

The anonymous survey shows that 75% of the participants agree or strongly agree that they learned how to use HomeGestures by observing another person. This raises the assumption that the effect of imitation plays an important role in these kinds of natural user interfaces. Nevertheless, observing alone is not sufficient, as 75% of the participants state that trying out the gestures by themselves was very important for their learning process. This evaluated aspect of the learning process provides interesting indications that should be further investigated as part of future work to provide statistical evidence. One of the implications of this finding would be to suggest more interactive tutorials, e.g. implying real-world demos or videos, when providing training material to future users of natural user interfaces.

6.2. NICE – Hands-free Natural User Interfaces in Smart Buildings

In this case study, we used the MIBO framework to build NICE (Natural Intuitive Camera-based Environment) control, a free-hand gestures and speech control for smart buildings [Schneider15]. It allows occupants to use freehand-gestures, speech commands and a multimodal combination of both to interact with a smart building. The system is based on a Microsoft Kinect v2 3D camera that senses the user's movements and speech commands. The usage of gestures and speech is a way to approach the challenge of interacting in an environment where everything is connected and addressable. NICE control allows the user to use gestures, speech or a combination of both modalities. Thereby, the user is em-

powered to interact with a variety of fixtures such as overhead lights, desk lights, blinds and any other device that is addressable and has an actuator. The interaction can take place from any position in the room as long as the camera can sense the user properly. As a result, the user is no longer required to walk to a light switch or to pull out a phone in order to interact with the environment.

Using the MIBO architecture, a significant set of framework features is reused. NICE serves as a sensor to detect gesture and speech commands independent from each other. MIBO integrates these multimodal inputs, applies the defined interaction model and links the raw input data to a meaningful command. This makes the configuration of interactions flexible and allows for quick evaluation iterations using real user tests. MIBO's feedback channel is used to provide feedback through any device that subscribes to the appropriate information, e.g. an Apple Watch.

In a user study, we evaluated the performance of the implemented gesture and speech commands with 16 participants. The performance of three modalities – gestures, speech and a multimodal combination of both – was compared for controlling lights and fans. Afterwards, the participants were asked to rank several aspects of the system. The study showed that people had the most fun using gestures to interact with the environment, and gestures are a valid and reliable approach for simple tasks. For complex tasks, participants considered the usage of a multimodal approach to be helpful. Visual and acoustic feedback is an effective means to reduce the number of failures, but can increase the required time to complete a task. In a self-experiment, two users have used the system over a longer period. Their increasing performance provides indications for the importance of learning, even for natural user interfaces.

6.2.1. Interaction Design

The wizard-of-oz study presented in section 3.2 has served as a basis to the interaction design of NICE. The gesture-based interaction uses a pointing gesture to select the fixtures to be controlled. The actual control command can be given using either another gesture or a speech command. Pointing was selected as an easy and natural way to refer and thereby select a fixture among several alternatives. The pointing is detected using stored positions of all fixtures and by calculating the probability of targeting a particular one. The position of the fixtures to be controlled is learned in the system using a triangulation-based approach. Thereby, a person points at a specific fixture from several positions in the room and the system calculates and stores the position of that fixture. Afterwards, the proba-

bility to point at a specific fixture is calculated using the hand and elbow position of the user's pointing arm and by calculating the distance of the line through these two points to all stored fixture positions. If this distance is low enough, considering factors such as the distance between the user and the fixture, the pointing is interpreted as selection. The two body joints, hand and elbow, were selected after experimenting with several alternatives, such as using the shoulder instead of the elbow. In order not to accidentally select a fixture within a movement, the pointing has to be stable for a specific amount of time to be registered. After a fixture was selected by pointing at it, its state can be switched (e.g. turning the device on or off) using the so called recurring-hand gesture. This gesture is performed by raising and then lowering the hand, returning to its original position. The elbow must be kept in about the same position during that movement. In order to avoid accidental detections, several additional constraints have to be fulfilled, e.g. a completion time, similar to the double click interval of a computer mouse. If the device aspect to be controlled is non-binary, such as a dimmable light, the gesture-based interaction consists of four parts. After the selection of the device using the pointing gesture, the continuous dimming process has to be started and stopped explicitly. This is achieved by raising the other, non-pointing arm above the head and lowering it again afterwards. After this movement has been performed, raising and lowering the pointing arm can be used to increase and decrease the light level. If the desired light level is reached, the dimming process has to be stopped by a defined gesture. This explicit stopping is required as else further movements of the pointing arm would be interpreted as commands to modify the light level. Speech commands have been enabled as an alternative way to start or stop the dimming process and to switch the state.

6.2.2. Implementation

The development of NICE was driven by three major design goals [Schneider15]:

1. **Decoupling from fixture controller.** NICE control is designed to be decoupled from the fixture controller that communicates with the fixtures. This is achieved by the integration with MIBO. While NICE is limited to detect gestures and speech commands and forwarding them to MIBO, the actual integration of multimodal input is performed in MIBO and then results in control commands to the fixture controller system.

2. **Loose coupling of input events.** Loose coupling of gestures and speech events al-

lows for the extensibility of new input events.

3. **Abstraction from image data.** Instead of working directly on image data, NICE control is designed to use the position of various human joints. This abstraction layer allows the adaption of NICE control to another camera or even a completely different recognition approach as long as joint position data can be gathered.

The resulting subsystem decomposition of NICE is shown in figure 6.4.

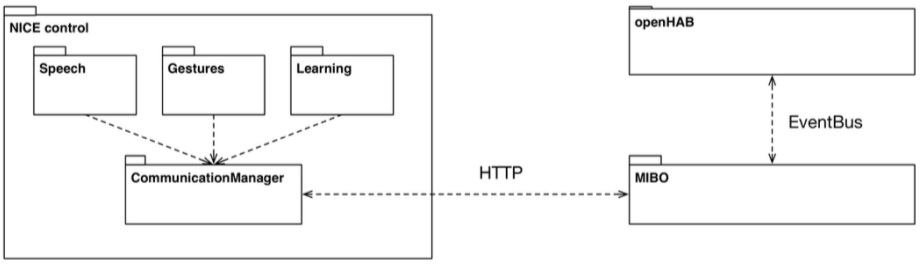


Figure 6.4.: Subsystem decomposition of NICE control. Source: [Schneider15].

To demonstrate the workflow within the participating systems that are involved to turn on a light, a sequence chart is shown in figure 6.5. The interaction starts with a user who performs a pointing gesture at a fixture. NICE constantly monitors the user's joint positions and detects the pointing. The pointing is evaluated and the precision is calculated. This information is forwarded to MIBO which evaluates and stores that information. Afterwards, the user performs a recurring hand gesture. Again, this gesture is detected by NICE control and forwarded to MIBO. MIBO evaluates that information and integrates it with the previous information. The particular interaction model describes that a pointing together with a recurring-hand-gesture should switch the state of the fixture that was pointed at. As a result, MIBO sends the command to turn on that light to the fixture controller (in our case *openHAB*) which turns on the actual light.

6.2.3. Related Work

The first work to combine several modalities in order to interact with the environment was shown by Bolt [Bolt80], where pointing was combined with speech input. Pointing is widely used as a means to select a device. Yamamoto et al. [Yamamoto04] showed an approach to detect intentional pointing gestures, but their system relies on multiple cameras and they only explained an application to control electronic appliances but did not

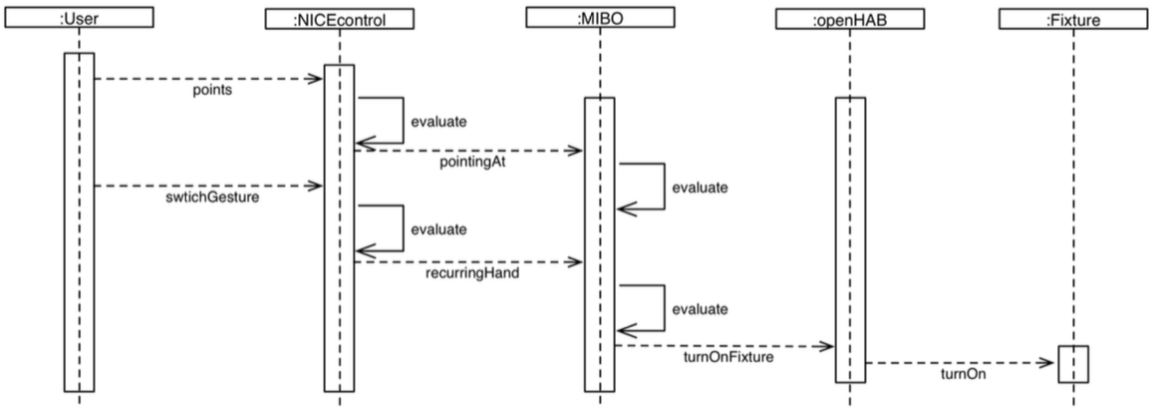


Figure 6.5.: Dynamic model of the user interacting with NICE. Source: [Schneider15].

evaluate the performance in such a setting. Caon et al. [Caon11] used multiple Kinect cameras to turn on or off devices in a smart room using pointing gestures. In contrast to this work, the pointing gesture itself switches states and dimming is not possible. In a project to support visually impaired people [ISSNIP13], the authors used a pointing gesture to detect the furnishings and objects in a room to allow visually impaired persons to get to know to a new environment. Compared to NICE control, this work only allows to select a device and tell the persons the devices that is pointed at, but an interaction with that device is not possible. Henze et al. [Henze10] showed a process to develop static and dynamic freehand gestures to control music playback with strong user involvement but they limited their scope to music playback. A hand-based gesture language was shown by Sadinejad et al. [Saidinejad14] who defined a vocabulary to control various parameters in a smart-home environment. While they interact with similar devices as in this work, they have specific gestures to select a device and do not use the location of the device. Besides performance and reliability, even social aspects play an important role for the design of a gestural language [Vaidyanathan14]. Ho and Weng [Ho13] presented favoured attributes for in-air gestures. They suggest natural initiations, such as gazing or gesturing towards an object to interact with it, and minimal effort as important implications for gesture design. Carrino et al. [Carrino11] showed a multimodal control for smart environments that uses image-, accelerometer- and microphone-data being captured using a handheld combination of a Wii Remote and a camera.

6.2.4. Evaluation

We conducted a user study to evaluate the performance of NICE. A total of 16 persons (eight male, eight female) from different origins participated in the study. The candidates have not used the system before. The study itself consists of two parts. The first part includes 30 tasks to interact with a room in an open office environment. The second part includes a structured interview where the participants rank eight aspects of the system on a 10-point scale with the option to comment. Additionally, we compared the performance to ourselves (two persons) who have been using the system throughout its four-months development.

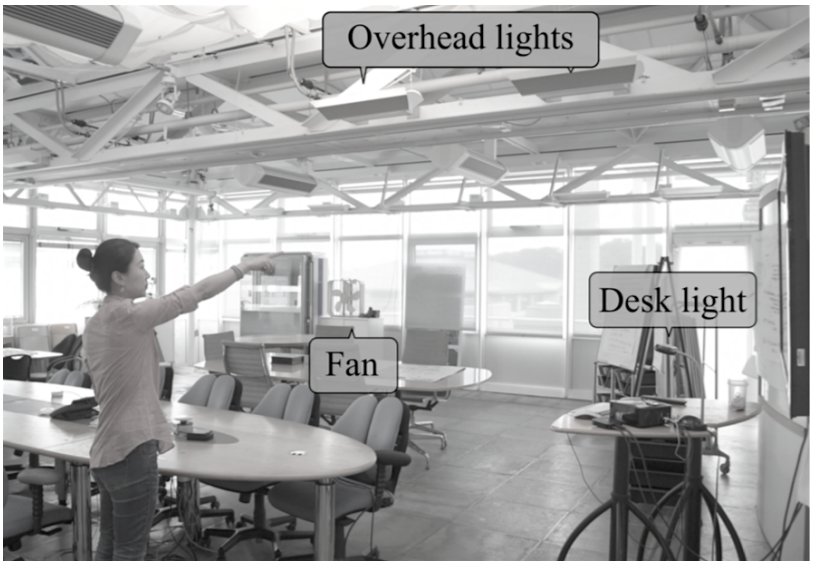


Figure 6.6.: Setup of the room and devices to be controlled. Source: [Schneider15].

For the first part, the participants were introduced to NICE control and the room (see figure 6.6) and have been shown the gestures they were supposed to use. Every task the participants were supposed to perform was explained on a screen in front of the participants, including the exact speech commands. The 30 tasks consisted of two sets of the same 15 tasks, one time performed with audio-visual feedback and one time performed without. Half of the participants started with feedback, the other half started without feedback. Two types of feedback supported the execution of tasks. An Apple Watch (see figure 6.7) showed the fixture that the user pointed at (or none if the pointing was not precise enough). Acoustic feedback in the form of a beep sound indicated that a speech command was detected and when a dimming process was started or stopped. The partici-

pants have been allowed to ask questions during the experiment if they were unsure about a specific task. The 15 tasks contained three sets with five tasks each. In each set, the participants have been asked to a use a specific modality - gestures, speech or the multimodal combination of both. The five tasks were the following:

- **T1** - Turn on an overhead light.
- **T2** - Increase light level for a second overhead light to 100% and dim the light afterwards.
- **T3** - Turn on a desk light.
- **T4** - Turn off the same desk light.
- **T5** - Turn on a fan (located further away).

While the tasks T1, T3, T4 and T5 all feature some kind of switching, T2 asked the participants to modify a continuous value. T2 represents a more complex task than turning something on or off.

Figure 6.7.: The visual feedback by an Apple Watch showing the current pointing.

Results

In total, 480 tasks have been analyzed. Out of these, 421 could be completed successfully, resulting in a failure rate of 10.6%. 14 failures occurred while using gestures, 8 while us-

ing speech commands and 29 while using a combination of both. The measured times in this study represent the duration from the point when the participant was asked to initiate the task to its successful completion. Overall, speech commands were the fastest (average times: 7.7s for speech commands, 12.6s for gestures and 11.2s for the combination of both) and required the least attempts to complete a task. However, in the overall ranking of the system from terrible (0) to perfect (10), participants ranked speech the lowest (6.3), followed by the combination (7.1) and gestures were ranked highest (7.4). Interestingly, even for the question to rank the interaction from slow to fast (10), speech was ranked lowest (5.9). An explanation could be that the expectation for the performance of speech commands is higher than for gestures, as the usage of speech for interaction is more common than the usage of freehand-gestures in everyday life.

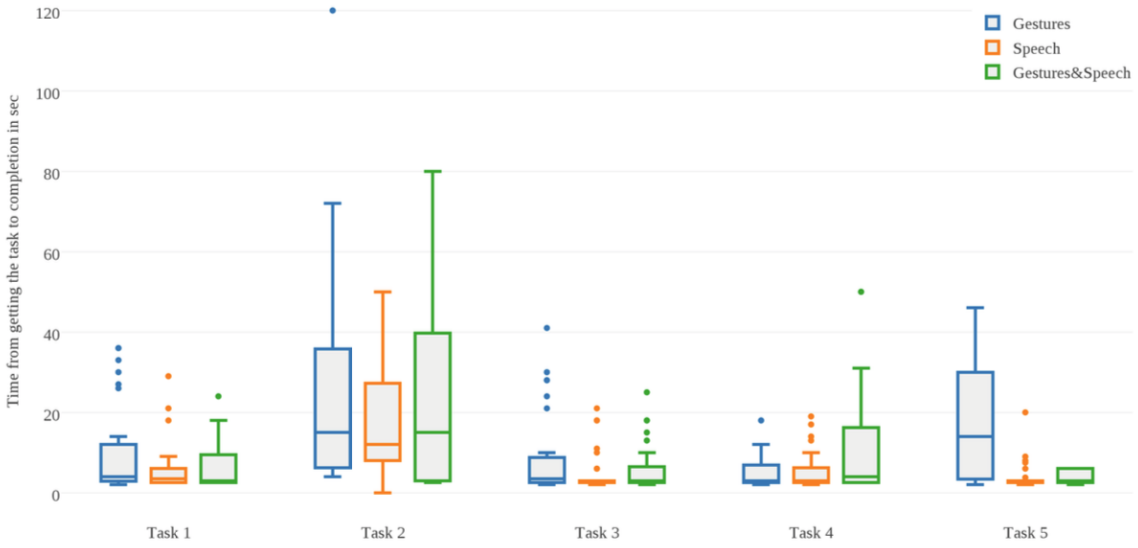


Figure 6.8.: Performance of successful tasks by modality, shown as boxplots. Source: [Schneider15].

12 out of the 14 failures using gestures occurred on the attempts to complete T5. The distance between the fan and the participant in the room was significantly larger than to the other fixtures. In general, T5 performed worse than other similar tasks, which leads to the assumption that the calibration of the position was bad. T1, T3 and T4 were always completed successfully for gestures and performed way better than T5. T5 was finished less than half as often at the first attempt than the other tasks. Using debug messages, we noticed that except for T5, the pointing itself worked mostly flawlessly, repeated attempts

or failures have usually been caused by problems during the recurring-hand gesture or the equivalent speech command. Sometimes, participants faced the camera in problematic angles and the human joint position data showed peculiar and wrong estimations that led to wrong or missing detections of gestures. The required timespan to complete a task decreased from T1 (10.2s) to T3 (8.8s) to T4 (4.9s) which indicates that people's performance increases after they learned how to use a specific gesture.

The speech recognition showed ambivalent performance. Beside huge differences among participants, the different commands showed large deviations in reliability. While commands such as *"Fan on"* worked flawlessly, other commands such as *"Start dimming"* were problematic. For some participants the system recognized the stop command *"Stop dimming"* all the time despite the participant saying *"Start dimming"*. This led to 13 failures for the multimodal dimming task, but despite its performance, participants ranked the combination of both modalities at 7.3 on a scale from annoying (0) to helpful (10). Especially for dimming tasks (ranked at 7.4 compared to 5.9 for switching tasks) people regarded the multimodal approach as supportive. Another example for the discrepancy between different speech commands can be seen in T3 and T4 in the multimodal case. Both tasks consisted of pointing at the desk light and afterwards giving the command to change the state to on or off. While pointing and saying *"Turn on"* only failed 3 times, pointing and saying *"Turn off"* failed 8 times and the successful attempts took in average about twice as long. In contrast, the context-aware speech command *"Turn it off again"* only failed once. T2 was proven to require a longer amount of time for all modalities compared to the other tasks (see figure 4). This was expected, as this task requires the selection of a device, starting a dimming process, modifying a value afterwards and finally stopping the dimming. This task features the longest times for completing a task and was the task with the most failures (excluding T5 for gestures) .

When we compare gestures for the group that started with feedback to the group that started without feedback, we notice that the first group was significantly slower for T5 (23.5s compared to 8.8s for successful tasks), but the success rate was also way higher (a third of the failures compared to the group starting without feedback). A reason for that might be that those starting with feedback figured out where they had to point to select the fan (T5) and were able to find it again later on. The group that started without feedback could only figure out where to point at to select the fan once they got the Apple Watch for visual feedback in their second part of the study. On the other hand, the group that started with feedback needed in average 14.7s to complete a task with gestures whereas

the group starting without needed about 10.5s. We see that the feedback mechanisms increase the success rate for interacting using gestures, but they also increase the time that is needed to complete a task. We assume the main reason is that people who started without feedback and had no problems pointing at fixtures did not feel the urge to look at the Apple Watch later on, while the group that started with feedback first tried to figure out whether the pointing is correct. The participants ranked the acoustic feedback at 8.6 and the visual feedback at 6.9 on a scale from pointless (0) to helpful (10). The lower ranking for the visual feedback might be caused by the additionally required time and by less problems performing pointings. The group starting with feedback found the visual feedback more helpful than the other group (7.4 compared to 6.4). There was no such difference for acoustic feedback. Most participants reported no effects of exhaustion with an average of 8.3 on a scale from terrible exhaustion (0) to no exhaustion at all (10). Only a few participants mentioned a slight exhaustion after gesturing throughout the experiment.

Participants have also been asked to rank the experience of interacting with the system between boring (0) and fun (10). The average score of 8.9 indicates most participants enjoyed this type of interaction with an environment. Especially gestures (ranked highest with 8.8) were appreciated as a means to interact. Speech was ranked lowest (6.7) for fun. Our hypothesis is that Apple's Siri and other intelligent personal assistants are commonly used and speech-based interaction is no longer novel and fun, as many people use such systems on a regular basis. The comparison of the 16 participants who used the system for the very first time to the two persons that used the system regularly throughout its 4-month development indicates that even natural user interfaces have to be learned. For the two experienced users, the performance was about twice as fast considering all modalities. About 73% of the interactions worked on the first attempt, compared to about 52% for the 16 participants. Similarly, the two experienced users needed in average 1.4 attempts to solve a task, compared to 1.9 attempts for the study participants.

Threats to Validity

It is important to notice that the implementation was based on the Microsoft Kinect v2 SDK in combination with the Microsoft Speech SDK for speech recognition capabilities. Especially in the combination with the integrated microphone of the Kinect v2, the Microsoft Speech SDK demonstrated poor performance which seems to be partly caused by too low volume levels. Some participants had serious problems with the speech recognition and were unable to get the system recognize their speech commands. In addition, the

command *"Turn on"* was often recognized as *"Fan on"*. As a result, T5 could be completed in some cases without a correct pointing in the multimodal case. The Kinect SDK showed wrong estimations of the position of several human joints if the person was either facing the Kinect directly or if the person was standing almost orthogonal towards the Kinect. Interestingly, despite these problems, there is no connection between poor performance of a modality and the rating of that modalitiy in the questionnaire afterwards.

Conclusion

Interacting in a smart environment by using free-hand gestures is a functional and fun way for occupants. For simple tasks and well calibrated positions of fixtures, pointing and a simple gesture was an effective means to control a fixture without any failures for all 16 participants. Participants enjoyed using gestures as a simple way to control fixtures in a room. The multimodal combination of gestures with speech was considered to be helpful, especially for more complex tasks such as dimming a light. The usage of speech commands seems to have lost its novelty, but performed best in this study. Interestingly, the perceived speed of speech commands was lower than for the other modalities, but the actual speed was highest. Participants preferred gestures or a multimodal combination over speech. This result supports the findings from our wizard-of-oz study (see section 3.2) showing that people would avoid speech especially in open office environments. Speed and reliability were the major concerns in this study, we hope for better speech recognition capabilities and for more reliable human joint position data in the future, especially when the user stands almost orthogonal towards the camera. Audio-visual feedback was considered to be helpful and led to a third less failures, but increased the time that was required to complete tasks.

Finally, MIBO's architecture has shown a multitude of advantages over a monolithic application, as the interaction model is decoupled from the recognition of the user input. This allows reusing of multimodal integration and feedback capabilities and allows for a flexible interaction design.

6.3. SISSI – Smart Interface for Speech Service Integration

With SISSI – a smart interface for speech service integration – we have shown the integration of MIBO with cloud-based speech services. This enables users to control a smart building using spoken natural language in a conversational style. Our wizard-of-oz exper-

iment in section 3.2 showed that speech has advantages over gesture-based controls when there is no spatial reference to single fixtures required. Especially, for supporting scenes (e.g. a presentation scene in a conference room or a reading scene in an office), speech is a convenient way to differentiate between numerous different settings. A scene represents a target state of a number of fixtures and thereby allows multiple fixtures to be set to a suitable state at once. In this case study, we demonstrate the usage of unimodal speech commands, complementing the existing MIBO framework with HomeGestures and NICE control. It has been tested in three buildings, American university offices, a German university and at the offices of a consulting company. These results are based on two preliminary studies, conducted by Breu [Breu12], and Henze and Peters [Henze16]. Both prototype implementations leverage Apple's speech assistant *Siri* to be used on iPhone, iPad and Apple Watch devices.

6.3.1. Interaction Design

SISSI enables conversational style commands, which allows users to speak in whole sentences. To extract a command from a spoken sentence, the commands should be separated into a *referent* (the target fixture for the command) and an *action* to be performed by a fixture. The action can be further specified by a *specifier*. Several commands can be grouped in a *scene*. The static object model in figure 6.9 shows these relationships.

The following subsections show examples of voice commands for controlling individual fixtures, collection of fixtures and scenes. When the user puts the action at the beginning of a sentence, the amount of possible words that can follow is reduced. Thus, the usage of this order results in a more reliable recognition. Optionally, the action can be further specified by a specifier, which allows to set particular values for the action (e.g. a light level). In order to identify the fixture that the command refers to, each command consists of a referent.

Individual Fixtures

The following list shows a selection of different commands to control individual fixtures split by actions (black), referents (light grey) and specifiers (dark grey). While the first six commands all relate to lights, the *"Turn on"* command can also be used to control other fixtures.

1. *"Turn the desk light on."*

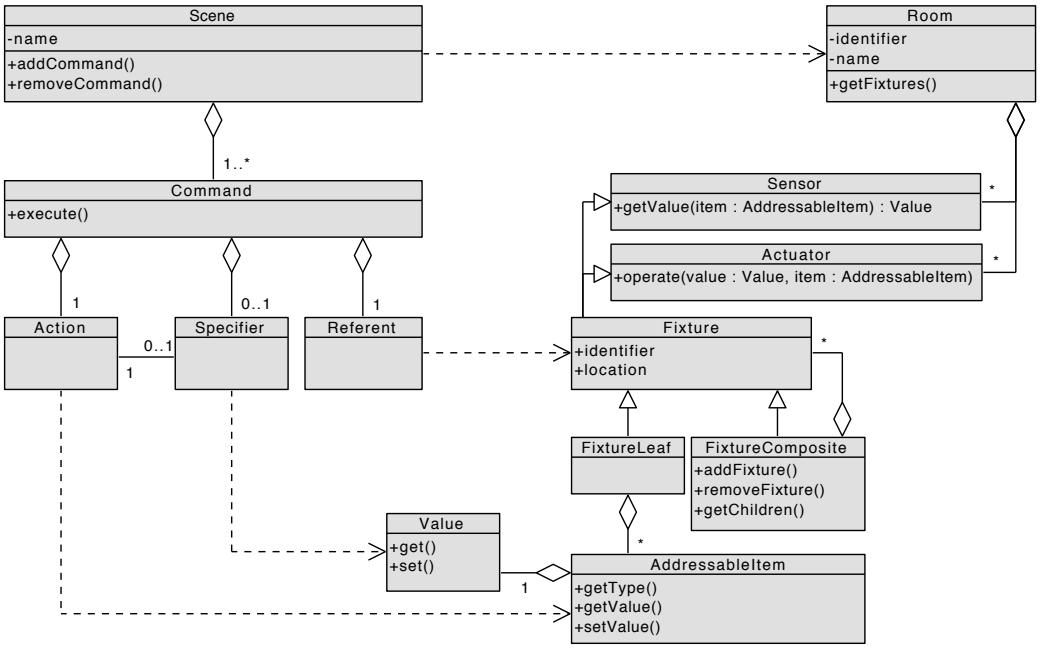


Figure 6.9.: Static object model of commands and scenes as UML class diagram.

2. *"Could you please dim the desk light for me?"*
3. *"Increase the brightness of the desk light by 10%, please."*
4. *"I would like to set the light level of the desk light to 50%."*
5. *"Please change the ambient light's color to red."*
6. *"On this beautiful day, please close the shades on the left."*
7. *"Close the shades on the right by 50%."*
8. *"Close the front door immediately, please."*

Collection of Fixtures

The second set of commands allows controlling collection of fixtures using a single command. A selection of typical commands is shown in the list below split by actions (black), referents (light grey) and specifiers (dark grey).

1. *"Turn on the lights in the conference room."*

2. *"Could you dim the lights in the conference room ?"*
3. *" Make the conference room the brightest , please."*
4. *"May I turn all lights on , please?"*
5. *" Set the temperature in my office to 20 degrees ."*

As shown in the wizard-of-oz study in section 3.2, controlling a collection of fixtures is an important aspect of the voice modality.

Scenes

A scene is a collection of commands that is executed at once. The following list shows examples of commands to activate a scene, split by actions (black) and referents (light grey).

1. *" Start the presentation ."*
2. *" Activate the presentation scene for me, please."*
3. *" Presentation scene , now."*

6.3.2. Implementation

The development of SISSI was driven by two major design goals:

1. **Decoupling of the speech-recognition.** The presented conversational style commands require more sophisticated approaches than plain word spotting techniques [Clapper71] [Christiansen76] for speech recognition. In addition, they require the incorporation of semantics (reflected in MIBO's interaction model) and context information. SISSI enables developers to integrate off-the-shelf speech assistants into their smart building controls by using adapters.

2. **Decoupling from modalities and fixture controller systems.** SISSI is designed to be decoupled from the fixture controller that communicates with the fixtures and from other modalities. This is achieved by the integration with MIBO. While SISSI is limited to receive speech commands from an off-the-shelf speech assistant and forwarding them to MIBO, the integration of multimodal input from other sources is performed in MIBO and then results in control commands to the fixture controller system.

The resulting subsystem decomposition is shown in figure 6.10. The `SISSIAdapter` connects the `SpeechAsssistant` with `MIBO` to allow for flexible multimodal interaction models. Thus, it abstracts the communication protocol of the `SpeechAssistant`, here the HomeKit Accessory Protocol (HAP), from the rest of the implementation. MIBO performs the integration of the speech modality with other modalities and applies the defined interaction model. As a result, control commands for fixtures are sent to the `FixtureController` which abstracts the communication with the fixtures in various protocols. The `UserInterface` is used to manage the room model, fixtures and the scenes that are stored in the `PersistenceManager`. To allow the `SISSIAdapter` to offer these services to different clients, SISSI follows the client-server architectural style.

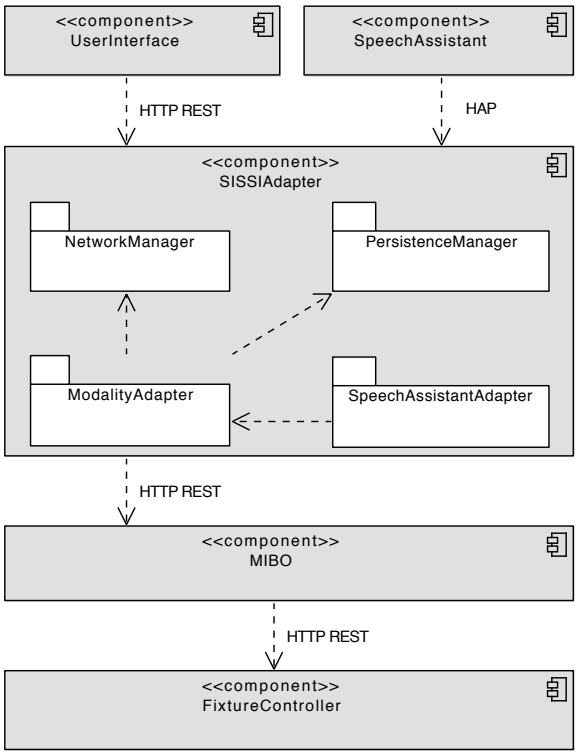


Figure 6.10.: Subsystem decomposition of SISSI.

SISSI has been tested in three buildings – American university offices, a German university and at the offices of a consulting company. Two prototypes have been implemented [Breu12][Henze16] based on Apple's speech assistant *Siri* to be used on iPhone, iPad and Apple Watch devices. Figure 6.11 shows an exemplary command of the voice control,

being operated from an iPhone.

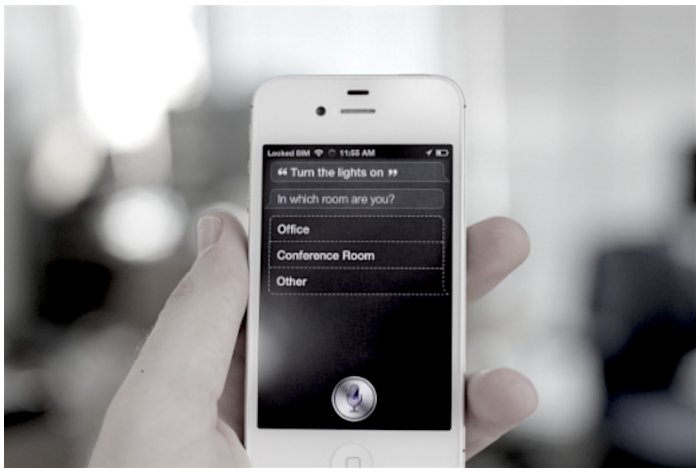


Figure 6.11.: Screenshot of Siri voice control. Source: [Breu12].

Context Awareness

To allow for intelligent actions, based on voice input, the incorporation of context information is of particular importance. For example, to allow for commands such as the following, the building must know about the user's position, the dedicated office, the current room situation, the persons surrounding the user and the personal calendar.

1. *"Turn the lights on in my office."*
2. *"Please remember this as work mode."*
3. *"I want to watch a movie."*
4. *"We are meeting in 20 minutes."*

We demonstrated two approaches to deal with user context. The fist approach uses a centralized repository storing context information as a directed acyclic graph (DAG) with a root [Breu12]. The second approach demonstrates distributed, personalized context information on each user's device [Henze16]. While the centralized approach shows advantages in a zero-enduser-configuration scenario, the second approach shows less privacy issues and simpler maintenance, e.g. to let the user change his/her office.

6.3.3. Related Work

Huggins-Daines et al. present PocketSphinx, a real-time continuous speech recognition system for mobile devices [Huggins-Daines06]. They ported and optimized Carnegie-Mellon's SPHINX-II, a popular open-source large vocabulary continuous speech recognition (LVCSR) system, to mobile devices. The resulting PocketSphinx system operates in average on a 206 MHz device, 8.03 times faster than the baseline system. The authors claim to be the first LVCSR system for mobile devices that is available under an open-source license.

Based on PocketSphinx by Huggins-Daines et al. [Huggins-Daines06], Barrena et al. showed a voice control for household devices [Barrena12]. Their target group consists of persons with neuromuscular disabilities. The system is based on speech recognition services on a smartphone and allows controlling a TV with seven commands (channel increase, channel down, volume increase, volume down, power, on and off). Both online and offline recognition were tested with variable distances between speaker and devices as well as different ambient noise levels. Successful recognition decreased significantly when increasing the distance from 0.5 meter to 1 meter. However, Barrena's work does not consider conversational-style speech commands, the user's context and the integration with multimodal controls.

Kaila et al. developed a location-aware speech control and audio feedback system for controlling smart environments [Kaila09]. Commands are recorded by an input device that consists of a main board, battery, FM radio transmitter, push button, microphone and infrared sensor. The transmission of audio starts when the user presses the push button. Users have to utilize direct commands, for example *"turn on lights"*. In this case, the system could respond with *"living room lights turned on"*. Kaila's control is location-aware by leveraging infrared emitters installed on the ceiling. The system was evaluated in the Smart Home laboratory at Tampere University of Technology, Finland. The results show that users were able to interact with the system. However, the test persons found it impractical to carry a separate device for speech recognition. Additionally, the positioning system faced issues when the view from device to the emitter was occluded.

Zhu et al. enhanced Kaila's approach and developed a voice control system for ZigBee-based home automation networks [Zhu10]. Their system uses SI-ASR (Speaker-Independent Automatic Speech Recognition) as voice recognition tool to avoid training of voice record-

ings and to simplify the initial setup. However, the necessity of carrying a microphone and the need of additional hardware components like ZigBee devices and infrared converters illustrate the disadvantages of this solution.

AlShu'eili et al. introduce a wireless home automation system (WHAS) with a voice-based user interface [AlShu'eili11]. They use ZigBee wireless communication modules to achieve a complete coverage of the controlled house. The speech recognition uses an external microphone and is implemented using the Microsoft speech API. It allows end-users to adjust the dictionary of supported voice commands. In their user study, 35 participants with different English accents performed 35 different voice commands. An average recognition of 79.8% could be achieved which results in five needed commands to achieve four given tasks.

Potamitis et al. introduce an integrated system for smart-home control of appliances based on remote speech interaction [Potamitis03]. Their system uses speech as a natural input modality to provide user-friendly access to information and entertainment devices installed in a smart home. Potamitis et al. focus on implementation details and practical considerations. In their work, they address the problem of hands-free speech recognition in a reverberant room with the presence of house-specific noisy conditions (e.g. TV/radio broadcast, interfering speakers, ventilator/air-condition noise). In order to reduce the amount of incoming signals to voices only, their microphone uses several filters, e.g. a band-pass filter to reduce the frequency spectrum. After these filters, a word spotter is used to search for predefined keywords and start the recognition process. The speech recognition uses a language dynamically created word lexicon. The lexicon was trained by 3000 speakers of a 5000 speakers voice database. This resulted in an average of 87% successful recognition for a given task on the first trial for three different scenarios.

In addition to scientific research, a multitude of voice-based user interfaces for smart buildings have been demonstrated in the commercial and hobbyist field. Appendix C provides an overview of these systems. However, none of these approaches shows the integration with multimodal building controls, a conversational-style for natural language and context-awareness.

6.3.4. Evaluation

We conducted two user studies to evaluate the performance of SISSI. In the first study, we tested with users who used SISSI for the first time. Afterwards, we provided a selection of five users with the opportunity to setup SISSI in their offices and on their own iPhone devices and use it in their daily work for two weeks. Afterwards, we conducted a second study with these users to evaluate the learnability of SISSI.

Study One: First Time Users

In our first user study, we tested SISSI with 20 users in order to evaluate the performance of the system and the acceptance with building occupants.

Study Design. Each user test was performed individually with the particular user. First, we introduced the participant to the setup of the fixtures to be controlled and the notion to use speech commands as if one would talk to another person. We gave each user nine tasks that people typically perform when they operate building controls. We did not provide the actual voice commands to be used. Instead, the participants had to come up with own ideas, how they would phrase the voice commands. From our participants, eight female and twelve male, the majority, 18 of 20, were not native English speakers and had different accents. The same iPhone was used by the participants to ensure the same settings. The users were asked to complete the following tasks:

- **T1** - Turn on a specific overhead light.
- **T2** - Turn off a specific overhead light.
- **T3** - Dim a specific overhead light to 80%.
- **T4** - Dim a specific overhead light to 0%.
- **T5** - Turn on all overhead lights in a room.
- **T6** - Turn off all overhead lights in a room.
- **T7** - Start a scene.
- **T8** - Turn off all lights in the building.

After the test, each user filled an anonymous survey. The first three questions were designed to gather information about the experiences of the participants and to confirm their expectations.

1. What would you like to control in a smart environment using voice?
2. What scenes would you like to have?
3. How often do you use Siri?

The next questions were asked about the voice control itself. While the first two questions were measured in a scale of one to ten to be comparable with the user study of NICE (see section 6.2), the remaining ten statements used a 5-point Likert-Scale [Likert32]. The following response category group was used: strongly disagree(1), disagree(2), neutral(3), agree(4), strongly agree(5).

1. Overall impression from terrible (1) to perfect (10).
2. The voice control is slow (1) to fast (10).
3. It is fun to control fixtures using voice.
4. I would feel more comfortable in my office with voice control.
5. Voice control will help me reducing energy.
6. Scenes provide significant additional value to controlling smart environments.
7. Siri speed to control devices is acceptable.
8. Voice control for smart buildings is faster than expected.
9. I would use Siri to control my devices.
10. Voice control will help me to retain control of the devices in my office.
11. I feel more comfortable to use Siri when other people are around me.
12. Voice control is more suitable to control a collection of devices than individual devices.

After the survey, we asked users to complete a last task (turn on an individual light fixture) where the system did not respond by intention. This task was used to measure the time until users repeat a given voice command if the system does not respond as expected.

	1 - Strongly Disagree	2 - Disagree	3 - Neutral	4 - Agree	5 - Strongly Agree	Total	Avg.
It is fun to control fixtures using voice.	0	0	1	11	8	20	4.35
I will feel more comfortable in my office with voice control.	0	2	6	10	2	20	3.60
Voice control will help me reducing energy.	0	2	4	11	3	20	3.75
Scenes provide significant additional value for controlling smart environments.	0	0	0	5	15	20	4.75
Siri speed to control devices is acceptable.	0	0	2	9	9	20	4.35
Voice control for smart environments is faster than expected.	0	0	3	8	9	20	4.30
I would use Siri to control my devices.	0	0	6	10	4	20	3.90
Voice control will help me to retain control of the devices in my office.	0	0	5	12	3	20	3.90
I feel comfortable to use Siri when other people are around me.	3	4	6	6	1	20	2.90
Voice control is more suitable to control a collection of devices than individual devices.	0	1	9	7	3	20	3.60

Table 6.3.: Questions and results of the first user study on SISSI. Source: [Henze16].

Results. Six participants never use Siri due to bad recognition performance or dialects and eight participants use it very rarely. Only six participants use voice control at least once a day. While the participants with prior Siri experience performed better in the first four tasks (avg. 60% less tries per task with Siri experience), the difference can be hardly seen in the last four tasks. This might indicate that learning to use voice control is fast and intuitive.

Out of all 160 observed commands (eight tasks with 20 participants), the participants needed an average of 1.78 tries per task when they used the system for the first time. While the average tries per task of T1 - T4 is at 2.34, the average of the T5 - T8 is at 1.21. This might indicate a better performance of voice control for collections of fixtures. However, since most of the participants rarely use Siri, the learning process during the tasks is another reasonable interpretation. The average wait time until the participants started to repeat a command was 5.51 seconds.

The overall impression reached an average of 7.50 out of 10 possible points, which is a higher than the 6.3 points for voice control in our NICE study (see section 6.2). A possible

explanation could be the better performance of the Siri speech recognition compared to the Microsoft Kinect SDK. 17 of 20 users agreed or strongly agreed that voice control for smart buildings is faster than expected. The other results of the survey can be seen in table 6.3.

Study Two: 14 Days of Usage

The second study was conducted to determine the performance of SISSI after users have used it for two weeks.

Study Design. The users from the first study who own an iPhone were asked to participate again to see changes in the results after 14 days of usage. To make sure that all participants are able to use the application, they participated in a short meeting in which the application was installed and explained. The six participants also had different accents but were fluent in English. Compared to the first user study, only little adjustments to the user study setup were made to ensure consistency of the studies and gain comparable results. The introducing questions were skipped as well as the final task for the latency test. Other than that, the tasks were the same and all other statements from the first user study were asked in the second study as well. In addition to the existing survey questions, the following seven questions were added to gather additional feedback from the experiment.

1. How often did you use Siri to control devices?
2. Voice control is useful.
3. Siri still needs a lot of improvement.
4. My inhibition level, when other people are around, dropped.
5. I will continue to use Siri to control devices.
6. The feedback provided by Siri is useful.
7. I am listening to the feedback provided by Siri.

While the statements 2-6 were asked using the Likert scale, the first and the last question had custom answer options.

	1 - Strongly Disagree	2 - Disagree	3 - Neutral	4 - Agree	5 - Strongly Agree	Total	Avg.
It is fun to control fixtures using voice.	0	0	1	3	2	6	4.17
Voice control is useful.	0	0	0	2	4	6	4.67
I will feel more comfortable in my office with voice control.	0	1	2	2	1	6	3.50
Voice control will help me reducing energy.	1	0	3	2	0	6	3.00
Scenes provide significant additional value for controlling smart environments.	0	0	0	3	3	6	4.50
Siri still needs a lot of improvement.	0	1	2	3	0	6	3.33
Siri speed to control devices is acceptable.	0	0	0	3	3	6	4.50
Voice control for smart environments is faster than expected.	0	0	2	3	1	6	3.83
I would use Siri to control my devices.	0	0	0	5	1	6	4.17
Voice control will help me to retain control of the devices in my office.	0	0	1	5	0	6	3.83
I feel comfortable to use Siri when other people are around me.	0	3	3	0	0	6	2.50
My inhibition level to use Siri, when other people are around, dropped.	0	1	3	1	1	6	3.33
I will continue to use Siri to control devices.	0	0	1	5	0	6	3.83
Voice control is more suitable to control a collection of devices than individual devices.	0	1	3	2	0	6	3.17
The feedback provided by Siri is useful.	0	1	1	3	1	6	3.67

Table 6.4.: Questions and results of the second user study on SISSI. Source: [Henze16].

Results. In this user study, a total of 40 tasks were performed with an average of 1.23 attempts per task. The tasks could only be performed by five of the six participants since one participant was unavailable due to sickness. With a result of 8.50 in a scale from terrible (1) to perfect (10), the participants rated the overall impression one point higher in the second user study than in the first one. Participants of the second study improved in all tasks regarding the average tries per task. All participants even improved to the optimum, one try per task, in six out of eight tasks. This performance is compared between both user studies in figure 6.12. The peaks in T3 and T4 are explainable due to a bad naming of the light fixture *"right light"*, which turned out to be tongue-twister for many users.

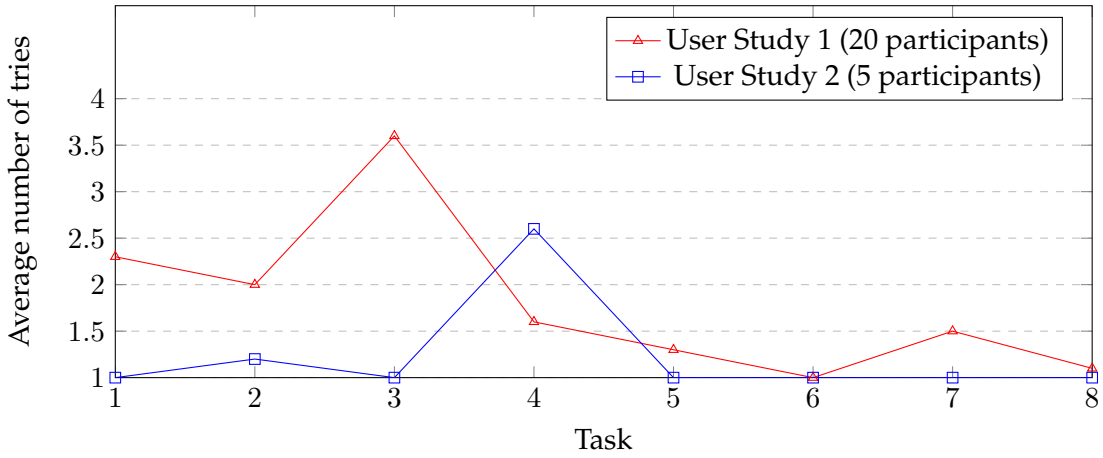


Figure 6.12.: Average number of tries per task

This emphasizes the importance of the naming of fixtures when implementing voice-based building controls. The other results of the survey can be seen in table 6.4.

Threats to Validity

Since SISSI relies on Siri for speech recognition, the success and failure of this study is highly related to the voice recognition quality of Siri. Poor speech recognition might decrease the rating at the overall impression. When participants have to repeat a voice command multiple times until Siri finally recognizes what the participant said, the impact on their fun might obviously be reduced. Additionally, with the majority of the participants not being native speakers, different accents led to misinterpretations by Siri. When comparing our two user studies, it is important to note that we conducted the second user study only with a subset of users from the first study (those who own an iPhone).

7. Conclusion

Buildings tend to rely on large zones of control [Loftness09], meaning that lighting, heating and ventilation are controlled for large areas with many occupants. This leads to two problems, discomfort for occupants and a waste of energy because each occupant may have different needs, depending on their tasks, clothing or habits [Park15]. Large control zones potentially tend to over-cool, over-heat and over-light entire areas, that may only be occupied by few people. Given the amount of time that people spend in buildings [BLS14], better indoor environmental quality (IEQ) can enhance the lives of building occupants. In particular, the ability to individually adjust parameters, such as heating, lighting, ventilation influences the IEQ and thus well-being, satisfaction and productivity of occupants [Gu11]. In offices, increasing the IEQ level and thus, the employees' health and productivity over the long run, can have a large return on investment, since personnel costs of salaries typically surpass building operating costs. Finally, given the significant share of buildings on the global energy equation [DOE13], energy savings in buildings are crucial to reducing our dependence on fossil fuels and greenhouse gas emissions.

We assume that providing individual control to building occupants is a key to comfort, well being and energy efficiency. MIBO's goal is to make building controls more intuitive to use, support the development process of control systems and provide a flexible platform to adapt interaction models at runtime. MIBO supports the integration of multiple modalities for natural user interfaces, e.g., gesture recognition, speech recognition, eye-tracking but also traditional button controls. Multimodal user interfaces (MMUI) are incorporated to make human-machine interfaces more similar to human-human interaction. MIBO provides an extensible domain-specific language to describe multimodal interactions in buildings and separate the interaction model from its implementation. This enables prototyping of new control systems for developers and allows users the configuration of desired interaction models to control fixtures in a building. The framework has been applied in three real-world applications, each in at least two different buildings with more than 20 users, demonstrating the feasibility and efficiency of our approach.

In this chapter, we summarize the contributions of this dissertation (section 7.1) and point out the limitations (section 7.2) of MIBO. Finally, we present a research agenda with emerging topics and questions from our work as part of future work (section 7.3).

7.1. Contributions

This dissertation provides four contributions to multimodal intuitive controls for smart buildings. First, we present a methodology for requirement elicitation of controls in smart environments to identify the modalities and suitable interactions within these modalities. Based on this methodology, we conduct real user experiments to identify requirements for intuitive building controls. Second, an extensible software architecture for multimodal integration and fusion is presented. Third, our tailored and extensible domain-specific language MiboML is defined to describe multimodal interactions in buildings and separate the interaction model from its implementation. Last, we showcase three building controls based on MIBO and evaluate them with real building occupants.

Methodology to find intuitively usable controls in smart environments

Our presented methodology to identify intuitively usable controls in smart environments combines bottom-up methods, i.e., wizard-of-oz and guessability study with top-down expert reviews. The combination of these methods increases the guessability of interactions and supports usability engineers in determining intuitive controls for smart environments. The guessability study gathers the users' intuitive ideas without forcing them to know the underlying system model. The wizard-of-oz technique mocks the system behavior and thereby adds a realistic interaction flow for the user. The top-down expert reviews pay special attention to aspects like the recognizability of interactions by the system to increase the fault tolerance. Additionally, feasibility for users on the long run is evaluated, such as the exhaustion effects of gestures.

Based on this methodology, we conducted real user experiments to identify requirements for intuitive controls in buildings. We asked test persons how they would control a smart room intuitively. The system was mocked by another human and responded appropriately to any given commands of the participants. Afterwards, the participants were questioned in a semi-structured interview. The results of our study support common as-

sumptions about the use of multimodal interaction within smart environments. The results point out the advantages of deictic gestures compared to speech for commands with a strong spatial reference, such as switching on a specific light fixture, and for simple commands when speech could be avoided. On the other hand, speech shows advantages for those commands without spatial reference, such as setting a temperature. Multimodal commands, incorporating both, speech and gestures, were most liked for more complex tasks. Functional and non-functional requirements are identified in the domain of intuitive building controls and compared to both, literature and our own studies.

Flexible software architecture for multimodal integration of smart building controls

The MIBO framework provides a reference architecture, consisting of a blackboard for the resolution of the multimodal input commands and the representation of knowledge as self-activating, asynchronous, parallel processes. It supports continuous interactions through iterative multimodal fusion with immediate user feedback. It fosters the use of software engineering principles, such as modularization, to support a growing set of input modalities as well as to enable the flexible configuration of interactions.

MiboML – A DSL for multimodal interaction models in smart buildings

MiboML is an extensible domain-specific language to describe multimodal interactions in buildings and separate the interaction model from its implementation. This allows for prototyping of new control systems by developers and allows end users to define and adapt the interaction model of a building. MiboML is a multimodal grammar that allows the instantiation of interaction models in buildings, represented by the MiboML syntax. An integrated development environment (IDE) can be used to define interaction models visually without programming skills. This allows occupants and facility managers of buildings to set up the required interactions and configurations of multimodal controls and adjust them to the occupant's needs.

Case studies of multimodal intuitive building controls

The use of the MIBO architecture and MiboML have been demonstrated in three case studies, a gesture-based smartphone control, a hands-free control using gestures and speech, and a speech-based control for smartphones and smartwatches. These case studies demonstrate the feasibility and efficiency of our approach in real-world applications, each in at

least two different buildings with different users.

The gesture-based smartphone control *HomeGestures* has been used for more than four years by over 20 users. Users simply point their smartphone at target objects and complete specific gestures. For example, pointing at the top of a window and making a down-gesture is interpreted as a command to lower the blinds at that window. Pointing the device at a light fixture and making an up-gesture raises the light levels. The implementation of the point-and-gesture control is based on the magnetometer, gyroscope and accelerometer built into recent smartphones. In combination with addressable fixtures and wireless infrastructures, this controller reveals how a wide variety of fixtures in a building can be controlled intuitively by pointing and gesturing. The evaluation shows that 94.1% of its users are having fun while controlling fixtures using HomeGestures. 82.3% of the users agree or strongly agree that HomeGestures helps to retain control of the fixtures in their office and that they feel more comfortable in their office with HomeGestures. 64.7% of the users agree or strongly agree that HomeGestures helps them reducing energy. We also investigated the question how people learn to use these kinds of natural user interfaces. We identified that three quarters of the users mainly learned how to use the gesture controls by observing another person and that trying out the gestures is important for users in the learning process.

We also used the MIBO framework to build *NICE (Natural Intuitive Camera-based Environment)* control, a free-hand gestures and speech control for smart buildings. It allows occupants to use freehand-gestures, speech commands and a multimodal combination of both to interact with a smart building. The system is based on a Microsoft Kinect v2 3D camera that senses the user's movements and speech commands. Thereby, the user is empowered to interact with a variety of fixtures such as overhead lights, desk lights, blinds and any other device that is addressable and has an actuator. NICE serves as a sensor to detect gesture and speech commands independent from each other. MIBO integrates these multimodal inputs, applies the defined interaction model and links the raw input data to a meaningful command. This makes the configuration of interactions flexible and allows for quick evaluation iterations using real user tests. MIBO's feedback channel is used to provide feedback through any device that subscribes to the appropriate information. In a user study, we evaluated the performance of the implemented gesture and speech commands with 16 participants. The study shows that people have the most fun using gestures

to interact with the environment and gestures are a valid and reliable approach for simple tasks. For complex tasks, participants consider the usage of a multimodal approach to be helpful. Visual and acoustic feedback is an effective means to reduce the number of failures, but can increase the required time to complete a task. In a self-experiment, two users have used the system over a longer period. Their increasing performance provides indications for the importance of learning, even for natural user interfaces.

With *SISSI* – a smart interface for speech service integration – we have shown the integration of MIBO with cloud-based speech services. This enables users to control a smart building using spoken natural language in a conversational style. Our wizard-of-oz experiment in section 3.2 shows that speech has advantages over gesture-based controls when there is no spatial reference to single fixtures required (e.g. by pointing). Especially, for supporting scenes (e.g. a presentation scene in a conference room or a reading scene in an office), speech is a convenient way to differentiate between numerous different settings. A scene represents a target state of a number of fixtures and thereby allows multiple fixtures to be set to a suitable state at once. In this case study, we demonstrate the usage of unimodal speech commands, complementing the existing MIBO framework with HomeGestures and NICE control. It has been tested in three buildings, American university offices, a German university and at the offices of a consulting company. The prototype implementations are based on Apple's speech assistant *Siri* to be used on iPhone, iPad and Apple Watch devices.

7.2. Limitations

We conducted formative evaluations with several smaller user groups instead of summative evaluations with a large group. This means that our evaluations are not based on statistically significant sample sizes. However, the approach of having smaller tests is supported by Nielsen [Nielsen00] who argues that the amount of gained insights decreases non-linearly with the number of users in a single test. The approach imposes the threat to validity that the results of the case studies would turn out differently with other user groups. In order to achieve statistical evidence, a larger amount of involved participants would be required.

Our case studies focus on typical building fixtures that are currently deployed in most

buildings, such as overhead lights, task lights, blinds, external louvers, thermostats and addressable plug loads. MIBO's concept of triggers and value providers may have to be extended for controls that require more complex interaction. However, MIBO has been designed as an extensible architecture to support future modalities and devices.

MiboML focuses on defining interactions for individual fixtures or a composite of fixtures (see page 65) with all fixtures controlled by the same command. For supporting a full scene concept, meaning that users can predefine states for multiple fixtures and activate them all at once, the meta model and language syntax must be extended to allow the definition of a set of predefined states for multiple fixtures when a certain command is triggered. However, since MiboML is highly extensible, this concept could be added without changing the overall system architecture. We have shown the integration of scenes with MIBO in our case study *SISSI* (see section 6.3). However, we did not use MiboML to define and configure the scenes in *SISSI*.

7.3. Future Work

Context reasoning. The focus of this dissertation is to demonstrate a framework with a reference architecture for the intuitive control of smart buildings. The blackboard is a flexible and suitable architectural style to allow for the incorporation of context information and reasoning, which has been shown in examples like the undo function for previously issued commands and the incorporation of the user's position using bluetooth beacon technology. Further incorporation of context information and reasoning remains part of future work. For example, when the *selector* of an interaction definition would not be provided by the user directly but rather a context knowledge source which resolves a command, e.g. "my daugther's favorite room", into an actual location. These context sources have to resolve ambiguous and imprecise commands. While it is expected by users that intelligent algorithms can perform this kind of reasoning, undeterministic commands are in fact problematic. Commands with an unpredictable outcome may cause safety issues in buildings and impose legal questions that are part of future work.

Futhermore, end-user programming might allow issuing and resolving commands such as "Close the blinds when I go to bed". The mechanism to resolve such commands is a trigger which is backed by one or multiple knowledge sources that combine multiple context dimensions to provide the information when a user goes to bed (e.g. time of day,

smartphone position, lighting in bedroom). These examples demonstrate the different aspects and the complexity of context awareness when controlling smart buildings. Each of the possible use cases requires a detailed consideration and enhancements with suitable knowledge sources for MIBO.

Recognition optimization. Our user study of the NICE control system in section 6.2 shows that false-positives, i.e. command is triggered unintentionally, and false-negatives, i.e. intentional command is not recognized, during the recognition process makes some interactions inadequate for certain situations. This can be due to the nature of the modality itself, as shown in our wizard-of-oz experiment in section 3.2 or by shortcomings in the recognition technologies. In the case of false-positives, MIBO offers effective methods to combine interaction definitions across devices, e.g. a watch has to be triggered before voice- or gesture-recognition is activated. In the case of false-positives, MIBO allows to enrich weak information with an additional modality, e.g. a weak pointing can be rendered more precisely by an additional voice command. However, the evaluation to improve each modality on its own with gestures and voice commands that are intuitive and recognizable at the same time has to be further investigated in future work.

Conflict resolution for building controls. In the context of this dissertation, we differentiate two kinds of conflicts that occur in the control of smart buildings, conflicting interaction definitions and conflicting user behavior. The first aspect is addressed in section 5.4. Interaction definitions may potentially obfuscate each other and trigger under the same conditions. We present a heuristic algorithm to detect these situations in advance.

The second type of conflicts may occur by several users giving contradicting commands to the same fixture or by activating contradicting scenes. This is a well known issue not only in smart buildings but also in traditional buildings, when fixtures are controlled by multiple switches. Conflict resolution strategies have to be developed in the future to resolve these issues. Solutions could embrace strategies for the prioritization of users and consensus-based approaches with averages or medians being negotiated by the system to maximize the occupants' satisfaction.

Pattern-recognition for repeating controls. MIBO offers potential to recognize building controls that are operated in a repeating pattern. This could lead to smart suggestions for automating fixtures and to consider this as context information for future interactions, e.g.

increase the probability for a certain interaction based on the context.

Test and simulation of interaction models. The testability of interaction models remains a challenge in MIBO. When an updated or newly created interaction model does not work as expected, troubleshooting may be hard for an end-user. A test mode for interaction models may be a solution that provides visual feedback in the MIBO IDE. The test mode could also show each parameter of an input and why it may not have triggered in a certain situation (e.g. "point-at duration too short?"). Furthermore, based on the validity time of hypotheses, a countdown could indicate the expiration time. This information would support users in adapting an interaction model more efficiently.

In addition to the simulation of interaction definitions, dependency injection would allow the creation of interaction models, even if not all recognition devices and modalities are physically available. For instance, if the current setup of a smart building does not provide a specific fixture or a system to recognize a dedicated modality, these elements could be simulated by mock objects being injected into MIBO. As a result, the system could transform more easily from a testing environment to a production environment and possible conflicts in definitions can be detected in advance.

Privacy. The issue of privacy becomes sensitive when ambient intelligence technology is used in people's private homes and in the office buildings that they spend significant times in. With the usage of mobile and wearable computing devices shadowing us at every turn, the incorporation of user context and location-based services points out the importance of confidentiality, anonymity, self-determination, freedom of expression and control of personal data. Proposed solutions often use various concepts for authentication, access control, privacy-by-design and data protection [Augusto10b]. Kinkelin et al. outline an approach for privacy preserving energy management in buildings [Kinkelin14]. They enforce strong access control using attribute-based cryptography and isolation of personal data using virtualization to ensure that users can only see information that they are actually interested in. While users could only see their personal energy consumption, an accountant could only see the total data of all users. The corresponding data is kept encrypted using private keys that reside at the individual users [Kinkelin14]. Fleck and Straßer show a privacy sensitive surveillance system for ambient assisted living [Fleck10]. Their approach is based on a camera sensor that transmits the results on a higher level of abstraction, e.g. "person injured" instead of the exact image data. The mentioned solu-

tions point out that security and privacy in smart environments, and their integration with usability, remains a complex research domain in future work.

Instrumented software applications. Just like instrumented buildings are equipped with sensors to learn about the conditions in the space, software applications can be extended with sensors as well. We consider these instrumented applications to be a subclass of instrumented environments, as defined in section 2.3. Roehm et al. [Roehm13] and Pagano et al. [Pagano12] have shown the usage of sensors in applications to monitor user interactions. These sensors, along with the recorded information, effectively support failure reproduction and software maintenance. Applying MIBO to these instrumented software applications remains to future work.

Smart cars. The number of employed sensors and actuators in cars has turned them into highly instrumented environments. Simple rules, such as turning on the windscreen wiper at rain or turning on the lights when it is dark, have been deployed since the 90's. Recently, however, the automotive industry has become increasingly interested in embedding sensing mechanisms that allow the car to make own decisions for a safer and less expensive journey [Augusto10b]. These *smart cars* include Cyber-Physical Systems (described in section 2.4) that actually communicate to systems in other cars ("Car-to-car communication") and make complex decisions, based on the context of the user and the environment. For example, a car which is involved in an accident may warn other cars on the same road to be prepared to brake. This information might be transmitted by cars passing the accident scene and going to the opposite direction.

Electric cars with their limited range compared to conventional cars have made the incorporation of the user context even more important, compared to conventional cars. For instance, the personal calendar of the driver might be queried to schedule the charging cycles and preheating/precooling of cars when they are connected to a power outlet. These measures can potentially extend the range of electric cars by more than 25% [Cooper16]. The incorporation of mobile technologies is closely related to this issue, as it provides complementary user control to initiate and schedule these kinds of services. Car manufacturers have successfully released several of these *connected car* services.

It remains to future work, to apply MIBO and multimodal interaction to other smart environments, such as smart cars.

Appendix

A. Glossary

The following section introduces definitions of ambiguous terms that are used continuously throughout this thesis in alphabetical order.

- (Control-) **Command.** A control command is issued by a user to change the state of a fixture in a building.
- **Fixture.** A fixture represents a controllable object in a smart building and affects the environmental conditions for an occupant. It is an instance of a specific fixture type.
- **Fixture Type.** A fixture type is the generalization of a fixture. Examples of typical fixture types in buildings: Overhead light, desk light, window blind, external louvers, operable window, coolwave, radiator.
- **Instrumented Environment.** An instrumented environment is an indoor environment with addressable sensors and actors.
- **Instrumented Space.** The term "instrumented space" is used interchangeably with "instrumented environment".
- **Interaction.** An interaction is defined by continuous or discrete user input in an arbitrary modality with the intend to issue a command.
- (Interaction-) **Definition.** An interaction definition, sometimes only called definition, describes one aspect of a MIBO interaction model. It defines one way to issue a control command for a particular fixture type, e.g. one way how users can dim a light in a building.
- **Interaction Model.** An interaction model describes the particular interactions that a user can perform to issue a control command for a fixture. It consists of many definitions, each describing one way to issue a control command for a defined scope of fixture types.

- **Modality.** A modality in the context of natural user interfaces "represent[s] information in some physical medium" [Bernsen08]. Each modality is characterized by its individual qualities and properties. The most relevant modalities in the context of this dissertation are described below:
  - Voice. Natural language is one of the most commonly known modalities to transport information. The semantics of the information is usually defined by the applied language. Additional information can be added by varying the volume or pitch level.
  - Gesture. A gesture involves any kind of body movement of a human-being. Typically, a movement of arms and hands is used to complete a gesture. However, any movement of the head, shoulders, or the upper body are considered to be a gesture as well.
  - Gaze. The movement and position of a human-being's eyes is considered as the gaze modality. Interactions like gazing at a particular position, winking or eye-rolling can be used to transport any kind of information.
  - WIMP. WIMP relates to Windows, Icons, Menus, Pointers, meaning a classical graphical user interface to transport information, such as using a keyboard, mouse or touch device and pressing buttons.
- **Occupant.** Any human-being residing temporarily or permanently inside a building is considered as an occupant of that building.
- **Smart Building.** A smart building is a smart environment which is embedded in a building.
- **Smart Environment.** A smart environment is an extension of an instrumented environment. It combines perceptual and reasoning capabilities with the other elements of ubiquitous computing to create a human-centered system that is embedded in physical spaces [Cook04]. A smart environment does not only provide some service automation, but furthermore learn and adapt its behavior during use [Augusto10b]. It is characterized by incorporating more complex reasoning capabilities to enable a more flexible and adaptive behavior, taking into account the context of the environment as well as the context of the user.
- **Smart Space.** The term "smart space" is used interchangeably with "smart environment".

B. History of Gesture-based User Interfaces

Research Study-year	Applications	Users-Gesture	Technology	Interface	Issues addressed	Result /Conclusion
"Put-That-There": Voice and Gesture[15]-1980	Pointing to items on the large screen with voice.	General-Hand	Llarge screen, a space-sensing cube on wrist, & microphone	Large screen display	How voice and gesture can be made to inter-orchestrate, actions	Conjoint use of voice- & gesture-recognition to command events on a large raster-scan display.
Hand drawn gesture for editing task [1]-1986	Text-editing	General-Hand	Stylus	Write on the surface of a display with a stylus	Consistency in using & gesture variability. Most common gestures Easy to use and remember	People behave in a way which makes gesture-driven interfaces feasible.
A hand gesture interface device[13]-1987	Manipulating pc objects using hand glove	General-Hand	Glove & the device incorporate a collection of technologies.	Computer screen with a 3D hand model, like mouse cursor	Manipulating 3D virtual objects with 2D controllers such as touch pads and mice are awkward.	Input devices, the Z-Glove and the DataGlove,
Charade[24]-1993	Presentation application using hand gloves	General-Hand	DataGlove with serial port of the pc.	Cursor movement in the computer screen using hand gesture.	Gestural command is represented by three icons. - distinguishes between gestural commands &other free-hand gestures.	Capture of gestures by using a DataGlove,
Camera based web interface by IBM[20]-1999	Web browsing with touch FREE switch	Disabled People-Body	Camera as input of web browser	Web interface	Real time behaviour of image processing for video. Mapping the gesture as command	Development of the TouchFREE applications.
Gesture Pendant [22] -2000	Controlling home appliances using wearable pendant	Disabled People-Hand	A small camera, part of a necklace, is ringed by IR LEDs & has a IR pass filter over lens, computer, controller device (slink-E, x10)	Control using wireless video	Used in a variety of lighting conditions. Enabling technology for elderly	Gesture controlled input device pendant can recognize 95% control gestures & 97% user defined gestures
GestureWrist and GesturePad.[19]-2001	Control any device using sensor and wired device in the wrist or Pad,	General -Hand	Input of wearable computer	In wrist and cloths. To control any device	Input devices should be natural and unnoticeable to use in various social settings.	2 Input devices for wearable computer.
Ubi-Finger[21] -2001	Control of appliance, presentation, window scrolling of Mobile PDA	General-Finger	Fingertip device with infrared sensor works via network.	Mobile PDA interface	Control various appliances by using existing metaphors and corporeality	Prototype systems, and evaluate how effective
XWand: UI for Intelligent Spaces[23] -2003	Interacting with intelligent environment using wireless sensor package	General-Hand	Handheld device , wand with variety of sensors & use Bayes networks	Natural interface with audio and LED feedback	Unifying the results of pointing detection and speech recognition,	Interface for intelligent environments

Figure B.1.: Source: Bhuiyan and Picking, 2009: Gesture-controlled user interfaces, what have we done and what's next? [Bhuiyan09]

Research Study-year	Applications	Users-Gesture	Technology	Interface	Issues addressed	Result /Conclusion
Visual Touchpad [6]-2004	Interaction with PCs using touchpad.	General-Hand	Quadrangle panel with a rigid backing with PCs & two cameras.	on PCs, large wall displays	Visualization of gesture in the screen. Singe and 2 handed gesture commands.	Vision-based input device that allows for fluid two-handed gestural interactions
Accelerometer based gesture control [4]-2004	Recognizing, training customised accelerometer based gestures	Elderly-Hand	Accelerometer, ubiquitous device	Accelerometer for recognition, training	Importance of the training in gesture control interface.	To reduce the error customization and training of gesture play important role
Gesture control for a design environment [3]-2005	Not specified.	Any-any	Study only	Study only	Usefulness of the gesture modality compared to other. Difference of gestures for the same task.	Gestures are a natural modality for certain tasks & can be augmented
Visualization method [2]-2006	Gesture visualization method, animation of hand movement performed during the gesture control	Elderly-Hand	Visualization method architecture using the accelerometer data	Gesture projection of the 3D path onto a plane, video, VRML	Concepts of the gesture visualization and it could be utilized in providing essential feedback and guidance.	Visualization provides information about the gesture performed.
http://atlasgloves.org -2006	Interaction with PCs using DIY hand gloves	General	Pair of illuminating gloves to track hand gestures. The Open Source Atlas Gloves application, webcam	Application in the PC	Controlling 3D mapping applications like Google Earth.	Open source API and DIY gloves
Intelligent Smart Home Control Using Body Gestures[11] -2006	control of smart home environments such as lights and curtains	General	Smart home with 3 CCD cameras. Marker attached in the human body	Gesture extracted as 3D and 2D view.	Continuously changing gesture can be used.	Recognition rate is 95.42% for continuously changing gestures
Head gesture recognition [7] -2007	Hands free control system of an intelligent wheelchair	Elderly & disabled	Wheelchair with laptop & webcam.	Laptop Interface with wheelchair user.	Gestures are used to generate motion control commands	Head gesture control system.
Select-and-Point[17] -2008	Enables controls to applications such as MS Office, web browser & multi-media programs in multiple devices.	General	Composed of three parts-a presence server, controlling peer & controlled peer using cameral, software tools	Table top, large screen PC or mobile	Eliminating different cumbersome processes in the group meeting & provides a intuitive interaction style through a pointing gesture	Implementing intelligent meeting room

Figure B.2.: Source: Bhuiyan and Picking, 2009: Gesture-controlled user interfaces, what have we done and what's next? [Bhuiyan09]

Name	Applications	Technology	Interface	Functionalities	Uses	Products
EYE TOY (Sony) -2003	Interacting as personalized gamer with Sony play station games.	Color USB digital camera device using gesture recognition and sound of microphone.	Game on TV, PC etc.	Uses computer vision & Gesture recognition to process images taken by the camera	Gaming application	Camera with console
Wii Nintendo[8] -2006	Wireless and motion-sensitive remote with game console	Game with any TV, computer etc.	Interface in the screen	Remote offers an intuitive, natural way to play games.	Gaming application control	Game console and remote control
XBOX live vision -2006	Interacting as personalized gamer with Microsoft's XBOX-360 games.	Color USB digital camera device using gesture recognition and sound of microphone.	Game on tv, pc etc.	Uses computer vision & Gesture recognition to process images taken by the camera	Gaming application	Camera with console
http://www.gesturetek.com/[25] -2008	Controlling pc, mobile or console application using camera or phone.	3D Camera for computer vision. camera in mobile device. pointing over frame	Mobile , pc, large screen table top interface.	Capturing gestures for normal PC operation. body gesture based gaming in the mobile device	More intuitive vision activities Without keyboard & mouse	Natural input for different media, PC, mobile, screen etc.
/www.mgestyk.com -2009	Interaction with computer to operate games and application	Software for hand-gesture processing & 3D camera	PC based interface	Capture hand movements and translate them into commands for controlling Windows application	Control games and other windows applications	Camera and software

Figure B.3.: Source: Bhuiyan and Picking, 2009: Gesture-controlled user interfaces, what have we done and what's next? [Bhuiyan09]

C. Voice-based User Interfaces for Smart Buildings

Title	Demonstration	Description	Hands-Free	Platform	Demonstrated Commands
Ridiculously Automated Dorm Room	http://youtube.com/watch?v=6x1GkgbVP1I	Automated dorm room with custom appliances.	✓	Custom	"Sleep Mode", "Party Mode"
Siri and Crestron	http://youtube.com/watch?v=8wQIbZuUHMg	Connects Siri to the Crestron home automation system.	✗	Siri, Crestron	"Turn on home theater", "Change channel"
Text to Home Appliances	http://youtube.com/watch?v=B9PbhZy_xk0	Control home appliances by sending text messages to a web service.	✗	Siri, Insteon thermostat	"Tell ... to turn off the bedroom fan"
The Romantic Mode	http://youtube.com/watch?v=9Nx1cYcrBR0	Control multiple appliances to a predefined setting.	✗	Siri, X10 home automation system	"Make it romantic in here"
Siri-controlled Coffee Machine	http://youtube.com/watch?v=cHKk5ju3jVU	Control a coffee machine.	✗	Siri, AMX, custom coffee machine integration	"Siri, make me a coffee"
IRIS - Smart Home Control	http://youtube.com/watch?v=OifmOJsbw50	A personal assistant on a PC/tablet capable of controlling home appliances.	✓	BVC, X10, Active Home Pro	"How cold is it?", "Put the telly on"
Voice Viper	http://youtube.com/watch?v=z8ztvutQnHw	A separate device controlling a underlying home automation system	✓	Voice Viper	"Kitchen TV satellite one", "Office TV roku", "deck speakers FM radio"
Simple Custom Home Automation	http://youtube.com/watch?v=QtSi7FQLE0g	Custom built voice controlled home automation.	✓	VRBot module, Atmel Microcontroller	"Spot lamp", "Desk Lamp"
Android Voice Control	http://youtube.com/watch?v=k1rB_1TLjmw	Home automation system on an Android tablet with voice control	✓	Android, SemVox	"Turn off the lights in the living room"

Figure C.1.: Overview of commercial and hobbyist applications of voice-based user interfaces for smart buildings. Source: Breu [Breu12].

Bibliography

[Aarts99] Aarts, E.; Appelo, L. Ambient Intelligence: Thuisomgevingen van de Toekomst. *IT Monitor*. 1999, pp. 7–11.

[Abowd99] Abowd, G.D.; Dey, a.K.; Brown, P.J.; Davies, N.; Smith, M.E.; Steggles, P.; Session, P. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*. Springer. 1999, pp. 304–307.

[Adler07] Adler, M.; Reynolds, F. Vision-Guided "Point and Click" for Smart Rooms. In *Second International Conference on Systems and Networks Communications*. IEEE. 2007, pp. 30–30.

[Afshin11] Afshin, A.E.; Akan, B.; Çürüklu, B.; Asplund, L. A general framework for incremental processing of multimodal inputs. In *Proceedings of the ACM International Conference on Multimodal Interaction*. 2011, pp. 225–228.

[Aldrich03] Aldrich, F. Smart Homes: Past, Present and Future. In *Inside the smart home*, volume 13. Springer. 2003, pp. 17–39.

[AlShu'eili11] AlShu'eili, H.; Gupta, G.S.; Mukhopadhyay, S. Voice recognition based wireless home automation system. In *4th International Conference On Mechatronics*. 2011, pp. 1–6.

[Anastasiou13] Anastasiou, D.; Jian, C.; Stahl, C. A German-Chinese speech-gesture behavioural corpus of device control in a smart home. In *Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments*. 2013, pp. 1–6.

[Ashton09] Ashton, K. That 'Internet of Things' Thing. 2009. URL `http://www.rfidjournal.com/articles/view?4986` [Last Accessed: 2015-04-08]

[Augusto10a] Augusto, J.C. Ambient Intelligence: The Confluence of Ubiquitous/Pervasive Computing and Artificial Intelligence. In *Handbook of Ambient Intelligence and Smart Environments*. Springer, Boston, MA. 2010, pp. 213–234.

[Augusto10b] Augusto, J.C.; Nakashima, H.; Aghajan, H. *Handbook of Ambient Intelligence and Smart Environments*. Springer, Boston, MA. 2010.

[Ayoub13] Ayoub, J. Towards Net Zero Energy Solar Buildings. Technical report, International Energy Agency. 2013.

[Barrena12] Barrena, S.; Klotz, L.; Landes, V.; Page, A.; Sun, Y. Designing Android applications with both online and offline voice control of household devices. In *38th Annual Northeast Bioengineering Conference*. 2012, volume 5, pp. 319–320.

[Bauereiß13] Bauereiß, S.; Peters, S. Indoor positioning of mobile devices using ultrasonic positioning and the blackboard architectural pattern. Unpublished, Technische Universität München. 2013.

[Beheim15] Beheim, M.; Peters, S. Energy awareness in smart buildings through personalized real time feedback on mobile devices. Unpublished, Technische Universität München. 2015.

[Bengtsson00] Bengtsson, T.; Saito, O. *Population and Economy*. Oxford University Press, Oxford. 2000.

[Bernsen08] Bernsen, N.O. Multimodality Theory. In *Multimodal User Interfaces*. Springer, Heidelberg. 2008, pp. 5–29.

[Bhuiyan09] Bhuiyan, M.; Picking, R. Gesture-controlled user interfaces, what have we done and what's next? In *5th Collaborative Research Symposium on Security, E-Learning, Internet and Networking*. 2009, pp. 513–521.

[BLS14] BLS. American Time Use Survey. Technical report, Bureau of Labor Statistics. 2014.
URL `http://www.bls.gov/news.release/pdf/atus.pdf`

[Bolt80] Bolt, R.A. "Put-that-there". In *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*. 1980, pp. 262–270.

[Bray06] Bray, T.; Paoli, J.; Sperberg-McQueen, C.M.; Maler, E.; Yergeau, F.; Cowan, J. Extensible Markup Language (XML) 1.1 (Second Edition). Technical report, W3C. 2006.

[Breu12] Breu, K. A Framework for Voice-based, Context-aware Control of Smart Buildings. Unpublished, Technische Universität München. 2012.

[Brumitt00] Brumitt, B.; Cadiz, J. Let there be light: Comparing interfaces for homes of the future. *IEEE Personal Communications*. 2000, *volume 28*, pp. 35–43.

[Burkhardt14] Burkhardt, F.; Schröder, M.; Baggia, P.; Pelachaud, C.; Peter, C.; Zovato, E. Emotion Markup Language. Technical report, W3C. 2014.

[Buschmann96] Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Michael Stal. *Pattern-oriented Software Architecture - A System of Patterns*. John Wiley & Sons Ltd., New York, NY. 1996.

[Cafaro14] Cafaro, F.; Lyons, L.; Kang, R.; Radinsky, J.; Roberts, J.; Vogt, K. Framed Guessability: Using embodied allegories to increase user agreement on gesture sets. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*. 2014, pp. 197–204.

[Caon11] Caon, M.; Yong, Y.; Julien, T.; Mugellini, E.; Abou Khaled, O. Context-Aware 3D Gesture Interaction Based on Multiple Kinects. In *The First International Conference on Ambient Computing, Applications, Services and Technologies*. 2011, pp. 7–12.

[Carpenter92] Carpenter, B. *The Logic of Typed Feature Structures*. Cambridge University Press, New York, NY. 1992.

[Carrino11] Carrino, S.; Péclat, A.; Mugellini, E.; Abou Khaled, O.; Ingold, R. Humans and smart environments: a novel multimodal interaction approach. In *Proceedings of the 13th International Conference on Multimodal Interfaces*. ACM. 2011, pp. 105–112.

[Chang08] Chang, J.; Bourguet, M.L. Usability framework for the design and evaluation of multimodal interaction. In *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 2*. British Computer Society. 2008, pp. 123–126.

[Choi10] Choi, J.H. CoBi: Bio-Sensing Building Mechanical System Controls for Sustainably Enhancing Individual Thermal Comfort. Ph.d. dissertation, Carnegie Mellon University. 2010.

[Christiansen76] Christiansen, R.W.; Rushforth, C.K. Word spotting in continuous speech using linear predictive coding. In *Acoustics, Speech, and Signal Processing*. 1976, pp. 557–560.

[Clapper71] Clapper, G.L. Automatic word recognition. *IEEE Spectrum*. 1971, *volume 8*(8), pp. 57–69.

[Cohen94] Cohen, P.; Cheyer, A.; Wang, M.; Baeg, S.C. An Open Agent Architecture. *AAAI Spring Symposium Series on Software Agents*. 1994, pp. 1–8.

[Cohen97] Cohen, P.R.; Johnston, M.; McGee, D.; Oviatt, S.; Pittman, J.; Smith, I.; Chen, L.; Clow, J. QuickSet: Multimodal Interaction for Distributed Applications. In *Proceedings of the Fifth ACM International Conference on Multimedia*. ACM. 1997, pp. 31–40.

[Cook04] Cook, D.J.; Das, S.K. *Smart Environments*. John Wiley & Sons, Inc., Hoboken, NJ. 2004.

[Cooper16] Cooper, C. EV EVERYWHERE: Maximizing Electric Cars' Range in Extreme Temperatures. Technical report, U.S. Department of Energy. 2016.

[Corno10] Corno, F.; Gale, A.; Majaranta, P.; Räihä, K.J. Eye-based Direct Interaction for Environmental Control in Heterogeneous Smart Environments. In *Handbook of Ambient Intelligence and Smart Environments*. Springer, Boston, MA. 2010, pp. 1117–1138.

[Coyle14] Coyle, T. Green Building 101: What is indoor environmental quality? Technical report, U.S. Green Building Council. 2014.

[D'Andrea09] D'Andrea, A.; D'Ulizia, A.; Ferri, F.; Grifoni, P. A multimodal pervasive framework for ambient assisted living. *Proceedings of the 2nd International Conference on Pervsive Technologies Related to Assistive Environments*. 2009, pp. 1–8.

[Darby06] Darby, S. The Effectiveness of Feedback on Energy Consumption. Technical report, Environmental Change Institute, University of Oxford. 2006.

[Daum11] Daum, D.; Haldi, F.; Morel, N. A personalized measure of thermal comfort for building controls. *Building and Environment*. 2011, *volume 46*(1), pp. 3–11.

[Dey00] Dey, A.K.; Abowd, G.D.; Salber, D. A Context-Based Infrastructure for Smart Environments. In *Managing Interactions in Smart Environments*. Springer, London, UK. 2000, pp. 114–128.

[DOE13] DOE. Buildings Energy Data Book. Technical report, U.S. Department of Energy. 2013.

[Dragicevic01] Dragicevic, P.; Fekete, J.D. Input Device Selection and Interaction Configuration with ICON. *People and Computers XV - Interaction without Frontiers*. 2001, pp. 543–558.

[Dumas09] Dumas, B.; Lalanne, D.; Oviatt, S. Multimodal interfaces: A survey of principles, models and frameworks. *Lecture Notes in Computer*

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2009, *volume 5440*, pp. 3–26.

[EPA89] EPA. Report to Congress on indoor air quality, Volume 2. Technical report, U.S. Environmental Protection Agency, Washington, DC. 1989.

[Erman80] Erman, L.D.; Hayes-Roth, F.; Lesser, V.R.; Reddy, D.R. The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty. *ACM Computing Surveys*. 1980, *volume 12*(2), pp. 213–253.

[Fernandes07] Fernandes, V.; Guerreiro, T.; Araújo, B.; Jorge, J.; Pereira, J. Extensible middleware framework for multimodal interfaces in distributed environments. In *Proceedings of the 9th International Conference on Multimodal Interfaces*. ACM Press. 2007, p. 216.

[Feuerstack12] Feuerstack, S.; Luís, R.W.; Carlos, S.; Paulo, S. MINT-Composer – A Toolchain for the Model-based Specification of Post-WIMP Interactors. In *XI Workshop on Tools and Applications, WebMedia*. 2012, pp. 3–6.

[Fischer08] Fischer, C. Feedback on household electricity consumption: A tool for saving energy? In *Energy Efficiency*. Springer. 2008, pp. 79–104.

[Fitts54] Fitts, P.M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. 1954, *volume 47*(6), pp. 381–391.

[Fleck10] Fleck, S.; Straßer, W. Privacy Sensitive Surveillance for Assisted Living – A Smart Camera Approach. In *Handbook of Ambient Intelligence and Smart Environments*. Springer, Boston, MA. 2010, pp. 985–1014.

[Flippo03] Flippo, F.; Krebs, A.; Marsic, I. A framework for rapid development of multimodal interfaces. In *Proceedings of the 5th international conference on Multimodal interfaces*. 2003, p. 109.

[Frankenberg16] Frankenberg, N.v.; Peters, S.; Brügge, B.; Loftness, V.; Aziz, A. Effective Visualization and Control of the Indoor Environmental Quality in Smart Buildings. In *Workshop Proceedings of the Software Engineering Workshops 2016*. 2016, pp. 124–129.

[Friedrich15] Friedrich, R.; Hiesel, P.; Peters, S.; Siewiorek, D.P.; Smailagic, A.; Brügge, B. Serious Games for Home-based Stroke Rehabilitation. *Studies in Health Technology and Informatics*. 2015, *volume 213*, pp. 157–160.

[Gal01] Gal, C.L.; Martin, J.; Lux, A.; Crowley, J.L. SmartOffice: Design of an intelligent environment. *IEEE Intelligent Systems*. 2001, *volume 16*(4), pp. 60–66.

[Gamma95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Boston, MA. 1995.

[Glasser11] Glasser, P.C. An introduction to the use of complementary filters for fusion of sensor data. Technical report, Glasser Communications. 2011.

[Gonzalez-Sanchez11] Gonzalez-Sanchez, J.; Chavez-Echeagaray, M.E.; Atkinson, R.; Burleson, W. Affective computing meets design patterns. In *Proceedings of the 16th European Conference on Pattern Languages of Programs*. ACM Press. 2011, pp. 1–11.

[Goransson03] Goransson, B.; Lif, M.; Gulliksen, J. Usability Design - Extending Rational Unified Process with a new discipline. *Interactive systems: design, specification and verification. 10th International Workshop*. 2003, pp. 316–330.

[Gu11] Gu, Y. The Impacts of Real-time Knowledge Based Personal Lighting Control on Energy Consumption, User Satisfaction and Task Performance in Offices. Ph.d. dissertation, Carnegie Mellon University. 2011.

[Hall97] Hall, D.D.L.; Member, S.; Llinas, J. An introduction to multisensor data fusion. *Proceedings of the IEEE*. 1997, *volume 85*(1), pp. 6–23.

[Hartkopf86] Hartkopf, V.; Loftness, V.; Mill, P. The Concept of Total Building Performance and Building Diagnostics. In G. David (Ed.), *Building Performance: Function, Preservation and Rehabilitation*. American Society for Testing and Materials, Philadelphia, PA. 1986, pp. 5–22.

[Hartkopf97] Hartkopf, V.; Loftness, V.; Mahdavi, A.; Lee, S.; Shankavaram, J. An integrated approach to design and engineering of intelligent buildings—The Intelligent Workplace at Carnegie Mellon University. *Automation in Construction*. 1997, *volume 6*(5-6), pp. 401–415.

[Henze10] Henze, N.; Löcken, A.; Boll, S.; Hesselmann, T.; Pielot, M. Free-hand gestures for music playback. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*. 2010, pp. 1–10.

[Henze16] Henze, D.; Peters, S. Voice Control for Multimodal Interaction in Smart Buildings. Unpublished, Technische Universität München. 2016.

[Hermann15] Hermann, M.; Pentek, T.; Otto, B. Design Principles for Industrie 4.0 Scenarios: A Literature Review. Technical report, Technische Universität Dortmund. 2015.

[Hersent12] Hersent, O.; Boswarthick, D.; Elloum, O. *The Internet of Things: Key Applications and Protocols*. John Wiley & Sons Ltd., Chichester, UK. 2012.

[Ho13] Ho, K.; Weng, H. Favoured Attributes of In-air Gestures in the Home Environment. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration*. 2013, pp. 171–174.

[Hoste11] Hoste, L.; Dumas, B.; Signer, B. Mudra: A Unified Multimodal Interaction Framework. In *Proceedings of the 13th international conference on multimodal interfaces*. 2011, p. 97.

[Huggins-Daines06] Huggins-Daines, D.; Kumar, M.; Chan, A.; Black, A.; Ravishankar, M.; Rudnicky, A. Pocketsphinx: A Free, Real-Time Continuous

Speech Recognition System for Hand-Held Devices. In *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*. 2006, volume 1, pp. 185–188.

[IFMA12] IFMA. Building Quality Assessment for Offices. Technical report, International Facility Management Association. 2012.

[ISSNIP13] ISSNIP. Real-time 3D pointing gesture with Kinect for object-based navigation by the visually impaired. *ISSNIP Biosignals and Biorobotics Conference, BRC*. 2013, pp. 1–6.

[Johanson10] Johanson, B.; Fox, A.; Winograd, T. The Stanford Interactive Workspaces Project. In *Designing User Friendly Augmented Work Environments*. Springer, London. 2010, pp. 31–61.

[Johanssen15] Johanssen, J.O.; Peters, S. An Interactive Editor for the Definition of Multimodal Controls in Smart Buildings. Unpublished, Technische Universität München. 2015.

[Johnston97] Johnston, M.; Cohen, P.R.P.; McGee, D.; Oviatt, S.L.; Pittman, J.a.; Smith, I. Unification-based multimodal integration. *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*. 1997, pp. 281–288.

[Johnston98a] Johnston, M. Multimodal Language Processing. *Center for Human-Computer Communication Oregon Graduate Institute*. 1998, pp. 2–5.

[Johnston98b] Johnston, M. Unification-based multimodal parsing. *Annual Meeting of the ACL*. 1998, p. 624.

[Johnston00] Johnston, M.; Bangalore, S. Finite-state multimodal parsing and understanding. *International Conference On Computational Linguistics*. 2000, p. 369.

[Johnston05] Johnston, M.; Bangalore, S. Finite-state multimodal integration and understanding. *Natural Language Engineering*. 2005, *volume 11*, pp. 159–187.

[Johnston09a] Johnston, M. Building multimodal applications with EMMA. In *Proceedings of the 2009 international conference on Multimodal interfaces*. 2009, pp. 47–54.

[Johnston09b] Johnston, M.; Baggia, P.; Burnett, D.C.; Carter, J.; Dahl, D.A.; McCobb, G.; Raggett, D. EMMA: Extensible MultiModal Annotation markup language. Technical report, W3C. 2009.

[Kaila09] Kaila, L.; Hyvonen, J.; Ritala, M.; Makinen, V.; Vanhala, J. Development of a location-aware speech control and audio feedback system. In *7th Annual IEEE International Conference on Pervasive Computing and Communications*. 2009, pp. 1–4.

[Kastner05] Kastner, W.; Neugschwandtner, G.; Soucek, S.; Newmann, H. Communication Systems for Building Automation and Control. *Proceedings of the IEEE*. 2005, *volume 93*(6), pp. 1178–1203.

[Kela05] Kela, J.; Korpipää, P.; Mäntyjärvi, J.; Kallio, S.; Savino, G.; Jozzo, L.; Marca, S.D. Accelerometer-based gesture control for a design environment. In *Personal and Ubiquitous Computing*. 2005, volume 10, pp. 285–299.

[Kelley84] Kelley, J.F. An Iterative Design Methodology for User-friendly Natural Language Office Information Applications. *ACM Trans. Inf. Syst.* 1984, *volume 2*(1), pp. 26–41.

[Kinkelin14] Kinkelin, H.; Maltitz, M.V.; Peter, B.; Kappler, C.; Niedermayer, H.; Carle, G. Privacy Preserving Energy Management. In *Proceeding of City Labs Workshop in conjunction with the International Conference on Social Informatics*. 2014, pp. 35–42.

[Koons93] Koons, D.B.; Sparrell, C.J.; Thorisson, K.R. Integrating Simultaneous Input from Speech, Gaze, and Hand Gesture. In M.T. Maybury (Ed.), *Intelligent Multimedia Interfaces*. American Association for Artificial Intelligence, Menlo Park, CA. 1993, pp. 257–276.

[Koß12] Koß, D.; Bytschkow, D.; Gupta, P.K.; Schätz, B.; Sellmayr, F.; Bauereiß, S. Establishing a Smart Grid Node Architecture and

Demonstrator in an Office Environment Using the SOA Approach. In *Proceedings of the First International Workshop on Software Engineering Challenges for the Smart Grid*. IEEE Press, Piscataway, NJ. 2012, pp. 8–14.

[Kumar00] Kumar, S.; Cohen, P.; Levesque, H. The adaptive agent architecture: achieving fault-tolerance using persistent broker teams. In *Proceedings of the Fourth International Conference on MultiAgent Systems*. IEEE. 2000, pp. 159–166.

[Larson09] Larson, J.; Raman, T.V.; Ragett, D.; Bodell, M.; Johnston, M.; Kumar, S.; Potter, S.; Waters, K. W3C Multimodal Interaction Framework. Technical report, W3C. 2009.

[Lawson09a] Lawson, J.Y.L.; Al-Akkad, A.A.; Vanderdonckt, J.; Macq, B. An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. *Symposium on Engineering Interactive Computing Systems*. 2009, pp. 245–254.

[Lawson09b] Lawson, J.Y.L.; Al-Akkad, A.A.; Vanderdonckt, J.; Macq, B. An open source workbench for prototyping multimodal interactions based on off-the-shelf heterogeneous components. In *Proceedings of the 1st ACM SIGCHI symposium on Engineering interactive computing systems - EICS '09*. 2009, October 2015, p. 245.

[Lazik12] Lazik, P.; Rowe, A. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. 2012, p. 99.

[Lebedev06] Lebedev, M.a.; Nicolelis, M.a.L. Brain-machine interfaces: past, present and future. *Trends in Neurosciences*. 2006, *volume 29*(9), pp. 536–546.

[Lee08] Lee, E. Cyber Physical Systems: Design Challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*. 2008, pp. 363–369.

[Likert32] Likert, R. A technique for the measurement of attitudes. *Archives of Psychology*. 1932, *volume 22*(140), p. 55.

[Liu09] Liu, J.; Zhong, L.; Wickramasuriya, J.; Vasudevan, V. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*. 2009, *volume 5*(6), pp. 657–675.

[Llinás11] Llinás, P.; García-Herranz, M.; Haya, P.A.; Montoro, G. Unifying events from multiple devices for interpreting user intentions through natural gestures. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2011, volume 6946, pp. 576–590.

[Lo10] Lo, K.W.K.; Tang, W.W.W.; Ngai, G.; Chan, S.C.F.; Tse, J.T.P. Introduction to a framework for multi-modal and tangible interaction. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*. 2010, pp. 3001–3007.

[Loftness09] Loftness, V.; Aziz, A.; Choi, J.; Kampschroer, K.; Powell, K.; Atkinson, M.; Heerwagen, J. The value of post-occupancy evaluation for building occupants and facility managers. *Intelligent Buildings International*. 2009, *volume 1*(4), pp. 249–268.

[López-Cózar10] López-Cózar, R.; Zoraida, C. Multimodal Dialogue for Ambient Intelligence and Smart Environments. In *Handbook of Ambient Intelligence and Smart Environments*. Springer, Boston, MA. 2010, pp. 559–579.

[Lou13] Lou, Y.; Wu, W. A Real-time Personalized Gesture Interaction System Using Wii Remote and Kinect for Tiled-Display Environment. In *25th International Conference on Software Engineering and Knowledge Engineering*. 2013, pp. 614–626.

[Lu12] Lu, J.n.; Qian, H.; Xiao, A.p.; Shi, M.w. Human-machine Interaction Based on Voice. In *AASRI Conference on Modelling, Identification and Control*. 2012, pp. 583–588.

[McGlashan04] McGlashan, S.; Burnett, D.C.; Carter, J.; Danielsen, P.; Ferrans, J.; Hunt, A.; Lucas, B.; Porter, B.; Rehor, K.; Tryphonas, S. Voice Extensible Markup Language. Technical report, W3C. 2004.

[McNamara08] McNamara, J. *GPS for Dummies*. John Wiley & Sons, Inc. 2008.

[Miller56] Miller, G.A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*. 1956, *volume 101*(2), pp. 343–352.

[Naumann07] Naumann, A.; Hurtienne, J.; Israel, J.H.; Mohs, C.; Kindsmüller, M.C.; Meyer, H.A.; Hußlein, S. Intuitive Use of User Interfaces: Defining a Vague Concept. In *Engineering psychology and cognitive ergonomics*. 2007, pp. 128–136.

[Neal91] Neal, J.G.; Shapiro, S.C. Intelligent Multi-media Interface Technology. In J.W. Sullivan; S.W. Tyler (Eds.), *Intelligent User Interfaces*. ACM, New York, NY. 1991, pp. 11–43.

[Nielsen00] Nielsen, J. Why You Only Need to Test With 5 Users. Technical report, Useit. 2000.

[Norman90] Norman, D.A. *The Design of Everyday Things*. Basic Books Inc., New York, NY. 1990.

[Nosovic14] Nosovic, S.; Peters, S.; Bruegge, B. Design of a Framework for Controlling Smart Environments. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. 2014, volume 8638, pp. 29–39.

[Novick06] Novick, D.G.; Ward, K. Why Don't People Read the Manual ? *Proceedings of the 24th annual conference on Design of communication*. 2006, pp. 11–18.

[O'Connor04] O'Connor, J. Survey on actual service lives for North American buildings. In *Woodframe Housing Durability and Disaster Issues*. Las Vegas, NV. 2004, October, pp. 1–9.

[OMG06] OMG. Meta Object Facility (MOF) Core Specification. Technical report, Object Management Group. 2006.
URL `http://www.omg.org/spec/MOF/2.0/`

[Oviatt97] Oviatt, S. Mulitmodal Interactive Maps: Designing for Human Performance. *Human-Computer Interaction*. 1997, *volume 12*(1), pp. 93–129.

[Oviatt02] Oviatt, S. Multimodal Interfaces. In *Handbook of Human-Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ. 2002, pp. 1–22.

[Oviatt03] Oviatt, S. Advances in robust multimodal interface design. *IEEE Computer Graphics and Applications*. 2003, *volume 23*(5), pp. 62–68.

[Pagano12] Pagano, D.; Juan, M.A.; Bagnato, A.; Roehm, T.; Bruegge, B.; Maalej, W. FastFix: Monitoring control for remote software maintenance. In *34th International Conference on Software Engineering (ICSE)*. 2012, pp. 1437–1438.

[Park15] Park, J.h. Are Humans Good Sensors? Using Occupants as Sensors for Indoor Environmental Quality Assessment and for Developing Thresholds that Matter. Ph.d. dissertation, Carnegie Mellon University. 2015.

[Pasat14] Pasat, S.; Peters, S. Collaborative Energy Saving in Smart Spaces Using Gamification. Unpublished, Technische Universität München. 2014.

[Pereira93] Pereira, F. Book Reviews: The Logic of Typed Feature Structures. *Computational Linguistics*. 1993, *volume 19*(3), pp. 544–552.

[Perroud11a] Perroud, D.; Angelini, L. Context-aware Multimodal Feedback in a Smart Environment. In *The First International Conference on Ambient Computing, Applications, Services and Technologies*. 2011, pp. 1–6.

[Perroud11b] Perroud, D.; Barras, F.; Pierroz, S.; Mugellini, E.; Khaled, O.A. Framework for Development of a Smart Environment: Conception and Use of the NAIF Framework. In *11th Annual International Conference on New Technologies of Distributed Systems*. IEEE. 2011, pp. 1–7.

[Peters11] Peters, S.; Loftness, V.; Hartkopf, V. The intuitive control of smart home and office environments. *Proceedings of the 10th SIGPLAN symposium on New ideas, new paradigms, and reflections on programming and software - ONWARD '11*. 2011, p. 113.

[Potamitis03] Potamitis, I.; Georgila, K.; Fakotakis, N.; Kokkinakis, G. An integrated system for smart-home control of appliances based on remote speech interaction. In *8th European Conference on Speech Communication and Technology*. 2003.

[Poyser13] Poyser, J. Making the Invisible Visible: The Psychology of Energy Use. Technical report, Indiana Living Green. 2013.

[Privat09] Privat, G.; Streitz, N.A. Ambient intelligence. In *The Universal Access Handbook*. CRC Press Taylor and Francis Group. 2009, pp. 60.1–60.17.

[Rajkumar10] Rajkumar, R.; Lee, I.L.I.; Sha, L.S.L.; Stankovic, J. Cyber-physical systems: The next computing revolution. In *Design Automation Conference (DAC), 2010 47th ACM/IEEE*. 2010, pp. 731–736.

[Reeves04] Reeves, L.M.; Martin, J.C.; McTear, M.; Raman, T.; Stanney, K.M.; Su, H.; Wang, Q.Y.; Lai, J.; Larson, J.a.; Oviatt, S.; Balaji, T.S.; Buisine, S.; Collings, P.; Cohen, P.; Kraal, B. Guidelines for multimodal user interface design. *Communications of the ACM*. 2004, *volume 47*(1), p. 57.

[Reithinger03] Reithinger, N.; Alexandersson, J.; Becker, T.; Blocher, A.; Engel, R.; Löckelt, M.; Müller, J.; Pfleger, N.; Poller, P.; Streit, M.; Tschernomas, V. SmartKom: adaptive and flexible multimodal access to multiple applications. In *ICMI '03: Proceedings of the 5th international conference on Multimodal interfaces*. 2003, pp. 101–108.

[Rieger05] Rieger, A.; Cissée, R.; Feuerstack, S.; Wohltorf, J.; Albayrak, S. An agent-based architecture for ubiquitous multimodal user interfaces. *Proceedings of the 2005 International Conference on Active Media Technology*. 2005, *volume 2005*(01), pp. 119–124.

[Roehm13] Roehm, T.; Gurbanova, N.; Bruegge, B.; Joubert, C.; Maalej, W. Monitoring user interactions for supporting failure reproduction. In *21st International Conference on Program Comprehension (ICPC)*. IEEE. 2013, pp. 73–82.

[Roscher09] Roscher, D.; Blumendorf, M.; Albayrak, S. Using meta user interfaces to control multimodal interaction in smart environments. In *Proceedings of the 14th international conference on Intelligent user interfaces*. 2009, pp. 481–482.

[Rosson02] Rosson, M.B.; Carroll, J.M. Scenario-Based Design. In *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*. L. Erlbaum Associates Inc., Hillsdale, NJ. 2002, pp. 1032–1050.

[Ruiz11] Ruiz, J.; Li, Y.; Lank, E. User-defined motion gestures for mobile interaction. *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. 2011, p. 197.

[Saffer08] Saffer, D. *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. O'Reilly Media, Inc. 2008.

[Saidinejad14] Saidinejad, H.; Veronese, F.; Comai, S.; Salice, F. Towards a Hand-Based Gestural Language for Smart-Home Control Using Hand Shapes and Dynamic Hand Movements. *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*. 2014, *volume 8867*, pp. 268–271.

[Sandor05] Sandor, C.; Klinker, G. A rapid prototyping software infrastructure for user interfaces in ubiquitous augmented reality. *Personal and Ubiquitous Computing*. 2005, *volume 9*(3), pp. 169–185.

[Schiele10] Schiele, G.; Handte, M.; Becker, C. Pervasive Computing Middleware. In *Handbook of Ambient Intelligence and Smart Environments*. Springer, Boston, MA. 2010, pp. 201–227.

[Schneider15] Schneider, A.; Peters, S. Natural User Interfaces for Controlling Smart Buildings. Unpublished, Technische Universität München. 2015.

[Schuler93] Schuler, D.; Namioka, A. (Eds.). *Participatory Design: Principles and Practices*. L. Erlbaum Associates Inc., Hillsdale, NJ. 1993.

[Serrano08] Serrano, M.; Nigay, L.; Lawson, J.Y.L.; Ramsay, A.; Murray-Smith, R.; Denef, S. The Openinterface Framework: A tool for multimodal interaction. *CHI '08 extended abstracts on Human factors in computing systems*. 2008, pp. 3501–3506.

[Sharma98] Sharma, R.; Pavlovic, V.I.; Huang, T.S. Toward Multimodal Human-Computer Interface. *Proceedings of the IEEE*. 1998, *volume 86*(5), pp. 853–869.

[Shin94] Shin, K.G.; Ramanathan, P. Real-time computing: A new discipline of computer science and engineering. *Proceedings of the IEEE*. 1994, *volume 82*(1), pp. 6–23.

[Simon07] Simon, R.; Fröhlich, P.; Obernberger, G.; Wittowetz, E. The Point to Discover GeoWand. In *International Conference on Ubiquitous Computing*. 2007, pp. 1–4.

[Stevenson13] Stevenson, A. (Ed.). *Oxford Dictionary of English*. Oxford University Press, 2nd edition. 2013.

[Streitz99] Streitz, N.A.; Geißler, J.; Holmer, T.; Konomi, S.; Müller-Tomfelde, C.; Reischl, W.; Rexroth, P.; Seitz, P.; Steinmetz, R.; Geibler, J.; Miiller-tomfelde, C. i-Land: An Interactive Landscape for Creativity and Innovation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1999, pp. 120–127.

[Sun06] Sun, Y.; Chen, F.; Shi, Y.D.; Chung, V. A novel method for multisensory data fusion in multimodal human computer interaction. In *Proceedings of OZCHI'06, the CHISIG Annual Conference on Human-Computer Interaction*. 2006, pp. 401–404.

[Teirikangas15] Teirikangas, J. HAVi : Home Audio Video Interoperability. Technical report, Helsinki University of Technology, Helsinki. 2015.

[Tóth09] Tóth, A.A.; Várkonyi-Kóczy, A.R. A hand gesture controlled interface for intelligent space applications. In *International Conference on Intelligent Engineering Systems, (INES)*. 2009, pp. 239–244.

[Tsukada04] Tsukada, K.; Yasumura, M. Ubi-finger: A simple gesture input device for mobile and ubiquitous environment. *Journal of Asian Information, Science and Life (AISL)*. 2004, *volume 2*(2), pp. 111–120.

[Turk14] Turk, M. Multimodal interaction: A review. *Pattern Recognition Letters*. 2014, *volume 36*(1), pp. 189–195.

[Ur14] Ur, B.; McManus, E.; Pak Yong Ho, M.; Littman, M.L. Practical trigger-action programming in the smart home. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. 2014, pp. 803–812.

[Vaidyanathan14] Vaidyanathan, V.; Rosenberg, D. "Will Use It, Because I Want to Look Cool" - A Comparative Study of Simple Computer Interactions Using Touchscreen and In-Air Hand Gestures. *Human-Computer Interaction Part II*. 2014, *volume 8511*, pp. 170–181.

[Watt11] Watt, S.M.; Underhill, T.; Chee, Y.M.; Franke, K.; Froumentin, Max Madhvanath, S.; Magaña, Jose-Antonio Pakosz, G.; Russell, G.; Selvaraj, M.; Seni, G.; Tremblay, C.; Yaeger, L. Ink Markup Language. Technical report, W3C. 2011.

[Wauchope94] Wauchope, K. Eucalyptus: Integrating Natural Language Input with a Graphical User Interface. Technical report, Naval Research Laboratory, Washington, DC. 1994.

[Weingarten10] Weingarten, F.; Blumendorf, M.; Albayrak, S. Towards multimodal interaction in smart home environments. *Proceedings of the 8th ACM Conference on Designing Interactive Systems - DIS '10*. 2010, p. 430.

[Weiser91] Weiser, M. The Computer for the 21st century. *Scientific American*. 1991, *volume 265*(3), pp. 94–104.

[WHO11] WHO. *World report on disability*. World Health Organization, Geneva, Switzerland. 2011.

[Wilson03] Wilson, A.; Shafer, S. XWand: UI for intelligent spaces. In *Proceedings of the conference on Human factors in computing systems - CHI '03*. ACM, New York, NY. 2003, pp. 545–552.

[Wobbrock05] Wobbrock, J.O.; Aung, H.H.; Rothrock, B.; Myers, B.A. Maximizing the guessability of symbolic input. *CHI '05 extended abstracts on Human factors in computing systems*. 2005, p. 1869.

[Wolpaw02] Wolpaw, J.R.; Wolpaw, J.R.; Birbaumer, N.; Birbaumer, N.; McFarland, D.J.; McFarland, D.J.; Pfurtscheller, G.; Pfurtscheller, G.; Vaughan, T.M.; Vaughan, T.M. Brain-computer interfaces for communication and control. *Clinical Neurophysiology: Official Journal of the International Federation of Clinical Neurophysiology*. 2002, *volume 113*(6), pp. 767–91.

[Wu10] Wu, J.; Pan, G.; Zhang, D.; Li, S.; Wu, Z. MagicPhone: Pointing & Interacting. *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing*. 2010, pp. 451–452.

[Yamamoto04] Yamamoto, Y.; Yoda, I.; Sakaue, K. Arm-pointing gesture interface using surrounded stereo cameras system. In *Proceedings - International Conference on Pattern Recognition*. 2004, volume 4, pp. 965–970.

[Yang07] Yang, S.e.; Do, J.h.; Jang, H.; Bien, Z.; Smith, M.; Salvendy, G. Human-friendly HCI method for the control of home appliance. In *Proceedings of the 2007 conference on Human interface: Part I*. 2007, volume 4557, pp. 218–226.

[York04] York, J.; Pendharkar, P.C. Human–computer interaction issues for mobile computing in a variable work context. *International Journal of Human-Computer Studies*. 2004, *volume 60*(5-6), pp. 771–797.

[Zelkha98] Zelkha, E. The future of information appliances and consumer devices. Technical report, Palo Alto Ventures. 1998.

[Zhu10] Zhu, J.; Gao, X.; Yang, Y.; Li, H.; Ai, Z.; Cui, X. Developing a voice control system for ZigBee-based home automation networks. In *2nd IEEE International Conference on Network Infrastructure and Digital Content*. 2010, pp. 737–741.