One-Shot Interactions with Intelligent Assistants in Unfamiliar Smart Spaces

by

Meghan Clark

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Associate Professor Prabal Dutta, Chair
Professor Mark W. Newman, Co-chair
Professor Marti Hearst
Professor Björn Hartmann

Summer 2021

One-Shot Interactions with Intelligent Assistants in Unfamiliar Smart Spaces

Abstract

One-Shot Interactions with Intelligent Assistants in Unfamiliar Smart Spaces

by

Meghan Clark

Doctor of Philosophy in Computer Science

University of California, Berkeley

Associate Professor Prabal Dutta, Chair

Professor Mark W. Newman, Co-chair

Smart space technologies have entered the mainstream home market. Most users currently interact with smart homes that they (or an acquaintance) have set up and know well. However, as these technologies spread to commercial or public environments, users will need to frequently interact with unfamiliar smart spaces. In such settings, users will be unaware of the available capabilities and the system maintainer will not be present to help. Users will need to quickly and independently 1) discover what is and is not possible, and 2) make use of available functionality. However, current smart spaces use a "proper device name" paradigm that requires users to know which individual devices are smart and what their proper names are in order to interact with the space. This approach makes it nearly impossible for users to operate unfamiliar smart spaces, especially when using natural language intelligent assistants. Widespread adoption of smart space systems, and all the benefits they may confer, will not be possible until this discoverability issue is solved.

In this work, we characterize the problems associated with the proper device name paradigm and evaluate potential solutions. First, we look at the limiting effect that device-centrism has on a user's conception of system capabilities, and find that framing interactions around intelligent assistants can mediate these negative effects. Then, we examine how the use of proper names to refer to system resources poses particular challenges in smart spaces. To address these issues while preserving an assistant framing, we borrow patterns from human-to-human messaging and apply them to human-to-assistant communication. Our method of using contextual photo messages enhanced by two technologies – augmented reality and autocomplete – allows users to determine available functionality and achieve their goals in one attempt with a smart space they have never seen before, something no existing interface supports. The ability to easily operate unfamiliar smart spaces improves the usability of existing systems and removes a significant obstacle to the vision of ubiquitous computing.

This having learned, thou hast attained the sum
Of wisdom; hope no higher, though all the stars
Thou knewest by name, and all the ethereal powers,
All secrets of the deep, all Nature's works,
Or works of God in Heaven, air, earth, or sea,
And all the riches of this world enjoyedst,
And all the rule, one empire; only add
Deeds to thy knowledge answerable; add faith,
Add virtue, patience, temperance; add love,
By name to come called charity, the soul
Of all the rest: then wilt thou not be loth
To leave this Paradise, but shalt possess
A Paradise within thee, happier far.

— Milton, *Paradise Lost*

To my family and friends.

Contents

List of Figures

List of Tables

Acknowledgments

The dissertation is supposed to draw a line of ownership through a body of research, presenting new human knowledge as the individual contribution of a single student who aspires to be recognized with a PhD in return. However, this narrative of the studious academic hermit is an illusion – while a single PhD student makes a dissertation, a whole lot of people make a PhD student. I owe much of my success in the PhD program to supportive and insightful advisors, colleagues, and friends and family over the years.

Any mastery I have of fundamental research skills can be credited to Prabal Dutta, my advisor at both the University of Michigan and UC Berkeley. Prabal taught me how to write, how to present, and how to ask the really important questions. He always let me run with whatever crazy idea I had at the time, and was supportive of my extracurricular activities and shenanigans – he ran interference for me with the authorities at least once! Prabal's express votes of confidence and his unflagging support established a steady foundation on which I could build my research career.

To help me navigate the field of human-computer interaction, I was fortunate to find a second adviser in Mark W. Newman, in the School of Information at University of Michigan. Mark patiently met with me weekly for years and helped me learn the vernacular of HCI. He also helped me find my people, because he is my people. The first time I looked at his dissertation, I discovered that it was eerily similar to early outlines that I had sketched out several years before, down to considering dropping out of the PhD to be a sheep farmer (I had been thinking goats, myself). While I have had fantastic advisors, Mark was one of the first professors that I saw myself in personally, and he made me feel like I belonged somewhere.

My dissertation and quals committee members Marti Hearst, Björn Hartmann, and Dan Klein have provided invaluable feedback and life advice over the last few years.

In my very first year, I was advised by Quentin Stout. I began my PhD expecting to do research in graph theory, but I never met my original advisor, as he went on paternity leave the summer before I even started. Quentin stepped up and took me under his wing. Our coffee conversations were a highlight of every week. He was there for me and made me feel cared for during a time in the program when it would have been easy for me to fall through the cracks.

An enormous amount of formative life experience during my PhD was due to Alex Halderman and the security group at University of Michigan. Alex brought me into the security family, where I went to very interesting places and met very interesting people.

For many years my Lab 11 labmates were my adopted family. Not only did we work together, we stayed up late into the nights, ate together, traveled together, celebrated together. Lab 11 taught me many things: 1) it's better to ask for forgiveness than permission, 2) every administrative deadline is optional, and 3) many hands make light work. Lab 11 gave me a sense of ownership and agency over everything that I hope to take with me wherever I go, from the physical space we inhabited, to departmental governance, to our research work. Lab inculcated a strong spirit when it comes to thinking about what is possible and how to do it

with flair. Lab 11 consists of incredible people, and I feel very lucky to have been part of such a special group.

I've had a number of interesting and formative conversations with members of the research community, from fellow students at Michigan and Berkeley, to fellow conference and workshop attendees. I thank you all! A few in particular stick out as having been particular impactful on the direction of the PhD, though they may not have realized it at the time: Rich Brown at LBNL, Anthony Rowe and Yuvraj Agarwal at CMU, and Ed Arens at UC Berkeley.

Many thanks and apologies are due to the wonderful administrators at University of Michigan and UC Berkeley. Kelly Cormier was particularly incredible so many times in defense of Lab 11 antics, and officially blessing the spiral staircase light installation and paying to bring it into compliance with the fire code. Shirley Salanio handled our transfer to UC Berkeley with speed, competence, and patience.

While at Michigan, I founded the CS Kickstart summer program. I like to send an enormous amount of thanks to my fellow CS Kickstart organizers, volunteers, and participants, as well as all of the experienced administrators who sat down in meetings with me and taught me how to get things done. In the nine months of near-constant panicked preparation, I learned so much, and felt so much fulfillment.

Friends are critical during the PhD, to celebrate the highs and get you through the lows. I'm loathe to enumerate people at the risk of forgetting someone, but here is a partial list: Eric Wustrow, who expanded my mind and taught me how to see things no one else sees, and do things that no one else would give you permission to do. My buddy Will Huang, who got me through the last several years of the PhD happy and mostly sane. Rohit Ramesh, who gets first pick of my books if anything happens to me. Liz Dreyer, Charlie Welch, Sai Gouravajhala, Dev Goyal, Daniel Galanos, Joe Maltby, Adwait Dongare, Miriam Goldstein, Adetunji Dahunsi, Mike Zandi, Dylan Cooper. Anne Mayoral and Remi Myers. My awesome brother James. My loving partner, David Bruns-Smith.

Above all, I owe the most thanks to my parents. Without them, this would never have happened. From my earliest days, Dad would stop at highway cut-throughs to show us fault geology, taught me the names of plants and trees, showed me how to listen to nature, and identify the stars. I would exaggerate illness so that Mom, a PhD student, would have to take me in to her oceanography lab instead of school. While working as a programmer at NASA, she taught me integer overflow and boolean logic. My parents proudly displayed books at the front of the door, and even when budgets were tight, I could always get a book. They both worked hard. I saw them retrain constantly, changing whole industries multiple times, balancing practical concerns with finding intellectual fulfillment. Moving to good school districts was their biggest priority. By example, they taught me the deep value of education. They also showed me what it looks like to put in a heroic amount of work hard to get yourself and your family to where you want to be.

Somewhere deep in the labyrinthine halls of my childhood memories there's a little girl holding her first microscope and draped in her mom's too-big labcoat, saying, "I'm going to be a scientist when I grow up!" After all these years, I can now finally tell her – we did!

Chapter 1

Introduction

Smart spaces are indoor spaces that contain *smart devices*—appliances and sensors that are connected, usually through networking technologies, to a computing fabric that renders them interactive. The idea of independently animated rooms or buildings has captured the public imagination for nearly a century, recurring frequently in movies, TV shows, books, and more. Smart spaces have frequently been used to probe our collective anxieties, but also excitement, about the current technological revolution, and especially its effects on domestic labor [1–3].

Early depictions of smart spaces, such as the smart home in the 1922 film "The Electric House" [4], emphasize automation through electromechanical means, and reflect contemporary hopes and fears about replacing human domestic workers with unthinking machinery as homes became electrified [1]. By World War II, middle-class Americans were no longer able to regularly hire servants, and housewives became the primary source of domestic labor [1, 5]. Consequently, the idea of a smart home became closely intertwined with the role of housewives. In 1967, Ruth Sutherland, the wife of the first smart home inventor, said, "At first, I thought it might really replace me! From the cartoons and jokes we see and hear about computers, isn't this the general impression that most homemakers at present would have if they suddenly found out they had a computer in their home?" [6].

In the 1980s, personal computers entered homes just as housewives were leaving them. In 1977, for the first time more than half of US women with children entered the labor force, a percentage that continued rising throughout the 1980s and 1990s until its peak at 74% in 2000 [1, 7]. Perhaps because of this, many depictions of smart homes during this time personify the home as a female caregiver, with a name and a voice, who is often explicitly cast in the role of a mother or housewife [3]. These "intelligent assistants" are able to observe what is going on in the space, and able to effect changes within it autonomously, usually with the goal of anticipating or fulfilling occupant desires.

In addition to exploring upheavals in domestic relations, smart homes have also been used as a lens on other social anxieties, helping us imagine a better future through technology [2]. They have been variously associated with energy efficiency in an age of climate change [8–11], protection against intruders [10–12], liberation from domestic chores [11], support while working [13], independence and safety for the growing population that is aging-in-place [10,

13–15], and companionship in the face of increasingly isolated living [16].

Innovators motivated by these compelling visions have at last taken smart homes from futuristic dreams to present reality. Technology has reached a point of maturity where many futuristic smart space features are within reach for a typical US consumer. Sensors and actuators have emerged on the mass market, providing the peripherals by which a DIY smart home can sense and act. These devices include motion, temperature, humidity, and light sensors, cameras, light bulbs, sprinkler systems, switches, thermostats, and speakers. Each manufacturer includes a smart phone app for manual remote control and sometimes automation of these devices [17]. A number of third-party platforms have also emerged that allow users to automate and connect compatible devices from different vendors [18]. Additionally, in recent years, commercial-grade intelligent assistants with the ability to integrate with these smart home devices have become resoundingly popular, with 46% of the US population reporting using assistants in 2017 [19]. While users primarily interact with intelligent assistants through smartphones [19], these assistants are also embodied in "smart speakers" and often act as points of voice control for a smart home [20], with over 80 million smart speakers sold thus far [21]. This brings us closer to the vision of the smart space as intelligent companion and helper who can frictionlessly understand and support our goals.

To-date, these smart space systems have mostly been retrofitted into homes by technology enthusiasts. A number of studies have found that smart homes are often maintained by a primary technical user, or "pilot user," but must also be used by "passenger users" such as roommates, spouses, or guests [22–27]. Though these passenger users are often not involved in the installation and configuration of the smart space system, they must nevertheless figure out what the system can do and how they can make the system do it. Currently, passenger users primarily discover available smart home functionality by relying on the pilot user to tell or show them [22].

However, as smart space technologies become more widely adopted outside the home, there will be an increasing number of passenger users in contexts where the system maintainer is largely inaccessible, such as commercial or public buildings. The current reliance on out-of-band social communication for discovering what devices are available and how to interact with them will become a significant obstacle to usability of smart spaces. In a space like a smart office or smart conference room, the building manager, facilities staff, or contractor who installed the system is unlikely to be present to inform the large number of passenger users how to use the system. These problems have already begun to emerge in residential situations where the smart home maintainer is not frequently present, such as the AirBnB home rental business (see Figure 1.1). Further, if the capabilities of the smart space change dynamically as mobile smart devices become increasingly common, not even the system maintainer may know what functionality is available at any given moment. Helping users quickly discover and invoke functionality in these unfamiliar smart spaces is critical if we wish to enable the widespread adoption of smart spaces, where every user is likely to be a passenger user at some point.

Supporting rapid capability discovery and invocation for users in unfamiliar smart spaces is a challenge due to the way we currently design smart space systems. Smart space systems are

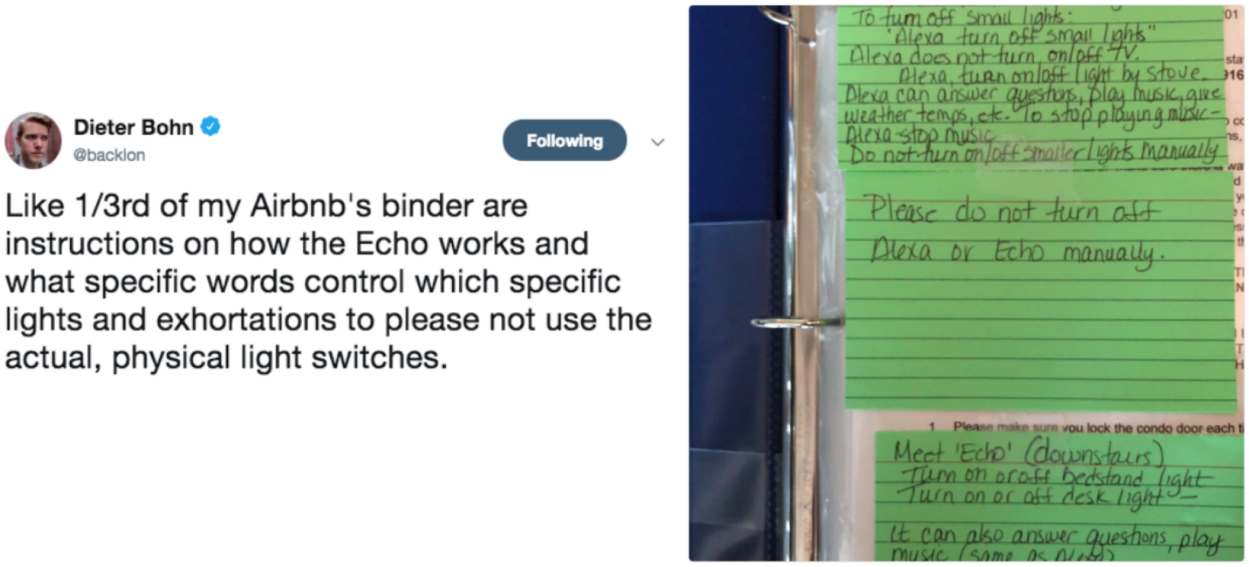


Figure 1.1: Ad-hoc discovery method for residential guests. The host must provide guests with the lexicon of proper device names. The host must also specify what the interface does *not* control (the TV), presumably because it is difficult to determine absence of support from failed interactions alone. Note that if users are unable to figure out the interface, they will manually turn off the devices, interrupting any automation routines. Reproduced with permission.

commonly presented as a collection of devices with proper names, and to invoke functionality, users must know the proper name of the relevant device—an approach we call the *proper device name* paradigm. However, there is little support for helping users discover the proper names of the space and the devices to which they correspond. Smartphone apps display identical-looking device icons labeled with proper names that may be hard to match to their real-world identities without using trial-and-error. Voice interfaces acutely lack discovery aids, providing few clues as to what the intelligent assistant understands [28]. The proper device name paradigm requires users to know *in advance* which devices are smart and what their proper names are in order to successfully interact with the smart space.

We show that while this approach may work for the users who set up the system and named the devices themselves, it does not scale to commercial buildings, hotels, offices, and public spaces where occupants cannot easily ask the system maintainer. We show that these leaky abstractions from the system-level conceptualization negatively impact widespread adoption by making user discovery fundamentally hard.

To characterize the problems caused by the proper device name paradigm, we run two separate studies. In the first study, we explore how *device-centrism* limits the what users expect a system can do, and show that an intelligent assistant approach is beneficial for smart homes in the long term. In the second study, we show how *proper names* are fundamentally not guessable and ambiguous in the smart space domain, further highlighting the need for

explicit discovery mechanisms.

While these two challenges are significant, the most popular smart space platforms are likely going to continue being built around the "proper device name" paradigm for the foreseeable future. Thus, we need a solution for passenger users to independently and quickly discover 1) what smart devices are available in a space and 2) what their names are. An interface that supports this form of discovery will not only help the user accomplish their immediate tasks, but may also help users learn information that can potentially transfer to other interfaces, unlocking their use.

Our goal is for users to walk into an unfamiliar smart space and accomplish a smart space task in a single attempt, or, if the task is not possible, not attempt to do it at all. We refer to this as a *one-shot interaction*. (This is not to be confused with the terms "one-shot learning" or "zero-shot learning" used in machine learning, which refer to entirely unrelated concepts.) Ultimately, we achieve our goal of supporting one-shot interactions with intelligent assistants in unfamiliar smart spaces, paving the way for truly ubiquitous computing.

1.1 The Emergence of Smart Spaces

Just as smart homes have been present in popular media for nearly a century, people have been trying to bring smart homes into reality since at least the 1960s. In 1966, a Westinghouse engineer named Jim Sutherland built a fully functioning smart home from spare computer parts that he called the ECHO IV, short for the Electronic Computing Home Operator [6]. The ECHO IV was the first system to prototype many of the functions we associate with smart homes today. The main controller was accessed through a nearby programming terminal, but there were also user-friendly interfaces distributed throughout the house, including a remote terminal in the kitchen, another control interface, and a controllable clock. While the administrative console needed to be programmed using an octal keypad, the other terminals allowed Sutherland's wife and children to interact with the home using a typewriter to support English and decimal numbers, allowing them to do tasks such as household budgeting, recipes and food inventory tracking, word processing, and interactive trivia games. Even as early as the 1960s, it was clear that passenger users wanted to interact with the home differently, and thus required a different kind of interface, than the system maintainer.

The first mass-marketed smart home was the Honeywell Kitchen Computer, advertised in 1969. The desk-sized kitchen computer was pitched as a way to help homemakers track recipes and pantry inventory [29, 30]. However, while the over $10,000 price tag included a one-week programming course for the primary user (who was assumed to be a housewife), the Honeywell Kitchen Computer was a wildly unusable device. The user had to translate ingredients into 16-bit codes, and then input those codes into the computer, which had no way of displaying information back to the user except for a series of LEDs, which would flash in particular patterns that the user had to interpret. Though it was advertised by Nieman Marcus, the Honeywell Kitchen Computer was not a commercial success [29].

From the 1960s through the 2000s, smart homes remained boutique systems accessible mainly to the wealthy, such as Bill Gates's residence as he described it in "The Road Ahead" [31]. However, this period also saw a blossoming of smart home research. At Xerox Parc, Mark Weiser's "The Computer for the 21st Century" launched the entire subfield of ubiquitous computing [32, 33]. The vision of small computing devices seamlessly integrated into the environment was a natural fit for smart homes, and several institutions built model smart homes for conducting research, including the Neural Network House at University of Colorado, Boulder [34], the Aware Home [35] and Gator Tech Smart House [36] at Georgia Tech, the Intelligent Room and House_n at MIT [37, 38], and the iRoom smart workspace at Stanford [39]. Researchers outlined the major research challenges [40–42], and a number of interfaces were developed, including Tangible Bits [43], the context-aware magnetic poetry (CAMP) interface [44], the interactive Context-aware Application Prototyper (iCAP) [45], and the Jigsaw editor [46, 47]. To support flexible smart space applications, middleware such as Speakeasy enabled interoperability between disparate devices [48].

While this proved to be a rich era for smart space research, the results did not directly translate into real-wold adoption. Even by 2010, the costs were too high for too few features, and too much technical expertise was required for most home occupants [9]. However, a confluence of forces has at last brought physical computation to the mass market.

1.1.1 The Internet of Things enters the home market

Recent years have seen a proliferation of low-power embedded sensors and actuators with networking capabilities, individually called "smart devices" and collectively dubbed "The Internet of Things" (IoT) [49–52]. The emergence of IoT technology was enabled by advances in a number of different fields. Transistor density increased while requiring less power, making it easier to perform useful computation on small, battery-powered devices [53, 54]. Battery technology itself improved, allowing smaller devices to operate for longer [55]. Cheap and low-power radios and networking protocols, such as Bluetooth Low Energy (BLE), Zigbee, and Thread, allowed a multiplicity of IoT devices to communicate with nearby gateways, while high-performance wireless home networks acted as backhauls [17, 51, 52]. Cloud technology enabled the formation of IoT businesses that could build value by aggregating data [51]. The ubiquity of the smartphone meant that most adults carried a network-connected user interface with them wherever they went, even in the home [17, 56].

While IoT technologies have been deployed in many sectors, including manufacturing, logistics, agriculture, and personal health, a significant domain for IoT devices is in smart spaces, where connected sensors and actuators primarily interact with home or office inhabitants.

In smart homes, as in the other domains, these IoT systems generally follow a "device-gateway-cloud" architecture, in which edge devices communicate using a low-power wireless protocol with a gateway, which then forwards any messages back and forth between the device and the cloud [17]. However, unlike the other IoT domains, non-technical users need to frequently interact with the system. This has led to a proliferation of user interfaces.

1.1.2 An ecosystem of smart space interfaces

The current landscape of interfaces has been influenced by two main factors: recent technology advances and the fractured device market.

The widespread adoption of smartphones and tablets beginning in the mid-2000s resulted in a glut of smartphone apps for the remote control of smart devices. Phone-based interaction methods have many advantages since they are a device that people already carry around with them, with a built-in display screen, touch input, and networking that is all handled by the phone already. This makes it easy for developers to make and for people to use smart device apps. It solves many of the problems of remote controls, in that the manufacturer only needs to write software rather than having to deliver a whole device, and the user does not need to go hunting for the remote since they will likely be carrying it around anyway.

The downside of having a separate smartphone app for each brand of device is the explosion of apps on the user's phone [17]. While setting up the system, the pilot user may be willing to install a number of different phone apps and remember which smart home device is controlled by which app, but such a burden becomes excessive for secondary and incidental users, such as spouses, roommates, and house guests. Consequently, third-party platforms emerged to unify devices from different brands into one universal interface, providing the convenience of a single rendezvous point in exchange for foregoing some brand-specific functionality. While some of these universal interfaces are yet another smart phone app, some are voice interfaces.

Natural language processing (NLP) also experienced a large leap forward in recent years driven by improvements in cloud computing and machine learning techniques [57]. Enabled by increasingly ubiquitous and high-bandwidth connectivity, speech interfaces on mobile devices began to send speech recordings to compute clusters in the cloud, which thanks to advances in hardware and software could run powerful new algorithms and respond with results in reasonable time [57, 58]. In 2017, Vaswani et al. introduced transformers, a highly connected neural architecture that uses attention to overcome the challenge of dependencies in sequences [59]. Combining transformers with a process for pre-training a large language model and then fine tuning it for specific applications led to Google's development of BERT [60], a very large pre-trained transformer model that advanced the state of the art in natural language processing. BERT was released as open source and needs little additional fine-tuning to achieve high performance on new applications, so it quickly became adopted by a many companies [57]. Modern systems have thus seen rapid improvements in voice recognition, natural language understanding of speakers' intents, and naturalistic speech synthesis. Subsequent large pre-trained models built on a transformer architecture, including GPT-3, promise more improvements ahead [61].

In smart spaces, these developments have resulted in the successful commercialization of intelligent assistants embodied by "smart speakers," microphone-and-speaker devices that allow occupants to talk to an intelligent assistant who can control smart home devices on the user's behalf. In 2020, nearly one-third of internet users had a smart speaker [21]. Of the estimated 83.1 million smart speakers worldwide, 69.7% allow users to converse with Amazon's Alexa, 31.7% embody Google Assistant, and less than 10% support Apple's Siri [21].

These different smart space interfaces have different strengths and weaknesses depending on the context. Device manufacturers' phone apps may support finer-grained controls, universal third-party apps may provide the best automation, and voice interfaces provide quick, hands-free operation. These interfaces act together as an ecosystem, allowing users with command over them to switch between different interfaces at various times depending on which is the most convenient in the current context and for their current goals.

1.2 Discoverability Challenges for Smart Spaces

Don Norman, author of the seminal work "The Design of Everyday Things," describes the discoverability of a system in terms of how well it supports users "figur[ing] out what actions are possible and where and how to perform them" [62]. When considering the usability of a system for new users, discoverability is often a key quality, since it is critical for supporting users who are just getting started with the system.

Discoverability starts before the user even sees the interface, with what we call the "interaction framing," or the conceptual model that the designer tries to convey to the user to help scaffold their understanding and expectations of the system. The interaction framing primes the user's mental model, shaping what they think the primary entities, states, and actions are. This can influence what interactions the user thinks is possible and what they will try to do with the system.

When using a smart space system, we observe that users must perform three kinds of discovery to determine what actions are possible and how to invoke them. We name these three semantic discovery, lexical discovery, and grounding discovery.

Semantic discovery involves learning what capabilities are (and are not) available, which is to say determining which devices in a room are (and are not) interactive and what states they can be in. For example, can the system control a particular light, and if so, can it change the light's colors?

Lexical discovery means discovering the vocabulary or keywords of the system that are required to invoke the functionality, particularly the proper names of the devices ("Left Reading Light," "Door"). Voice interface users may additionally need to discover special keywords referring to actions ("turn," "unlock"), attributes ("power," "brightness," "temperature"), or values ("on," "bright," "68 degrees Fahrenheit").

Grounding discovery means matching proper names to devices. Beyond just learning which devices are interactive and the set of available device names, users must know additionally which names correspond to which devices.

1.2.1 How do passive users currently perform discovery?

Multiple studies of smart home users have reported that users can be roughly categorized into tech-savvy users who install and maintain the system ("pilot users"), and secondary or incidental users ("passenger users"). Passenger users can be roommates, spouses, parents,

children, or guests [22–27]. Since pilot users usually set up and administer the system, they are intimately familiar with the system's capabilities and configuration by the time the system comes online. Passenger users, on the other hand, must discover the available capabilities and how to invoke them with little prior knowledge.

One survey of 178 smart home users found that passenger users primarily learn what they can do and how to do it by observing a pilot user's usage, or by explicitly asking a pilot user [22]. Additionally, passenger users tend to be uninterested in investing effort to learn the overall capabilities of the system, merely the feature they wish to use. However, this indicates trouble for the further adoption of smart spaces. As smart spaces spread into commercial and public environments, the pilot user may not be available to play this crucial discovery role when a passenger user enters an unfamiliar smart space (see Figure 1.1).

1.2.2 The rising importance of unfamiliar smart spaces

While residential occupants can fall back on social mechanisms to overcome discovery challenges, in commercial and public buildings, the lack of discoverability is a more significant obstacle. The informal social workarounds developed by users in residential settings do not work in commercial buildings where the smart device installation will likely be done by the building manager or a contractor. The vast majority of occupants in commercial buildings will not be familiar with what devices have been installed, what they have been named, or how users can interact with them. Additionally, many commercial buildings have a substantial number of visitors who will need to be able to discover how to control relevant devices (e.g., lights, projectors, and blinds) as quickly as possible.

As smart buildings continue to advance and mobile platforms like wearables and cars become integrated into the infrastructure, available system capabilities may change so often that even system maintainers may not know what specific services are available at any particular time. Successful interfaces will need mechanisms to dynamically reflect available capabilities in real time.

For smart spaces to grow in adoption, scale into commercial and public spaces, and support dynamically changing capabilities, smart space interfaces will need to be geared towards supporting the needs of users who wish to use the capabilities of unfamiliar smart spaces.

1.2.3 Problems with the proper device name paradigm

Independent user discovery is currently a challenge in smart spaces because the vast majority of smart space interfaces follow what we call the *proper device name* paradigm. These systems require that users learn and remember the proper names of devices *in advance* in order to invoke their functionality.

Smartphones apps exhibit this paradigm by representing real-world devices with icons that are identical within their device category (e.g., the same light bulb icon for various smart lights, the same switch icon for different brands of switches), differentiated only by their proper name. A passenger user scrolling through such a list, without prior knowledge from

the pilot user that set up the system, may need to use trial-and-error to determine which name belongs to the device they wish to operate. Voice interfaces result in an even more dire situation, as users may not even get the list of names or hints as to the device type. Passenger users must learn the proper name through some other method, and then recall from memory the proper name of the device they want to invoke.

While passenger users currently learn proper names by asking or observing the system maintainer, no interfaces exist that help users discover the information for themselves when the system maintainer is unavailable. In this work, we characterize how difficult it is for users to perform independent discovery in unfamiliar smart spaces built around a proper device name paradigm, and explore new interfaces to improve smart space discoverability.

1.3 Thesis Statement

In a future where smart spaces are ubiquitous in residential, commercial, and public buildings, users will need to frequently interact with unfamiliar smart spaces. To have successful interactions in an unfamiliar smart space, users must solve several discovery problems:

- What capabilities the space has (semantic discovery)
- What names the system understands (lexical discovery)
- Which names refer to what capabilities (grounding discovery)

Thesis: Framing a smart space as an intelligent assistant rather than as a collection of connected devices better conveys a sense of interoperability and automation to users. We can support these three kinds of discovery in unfamiliar smart spaces while preserving an assistant-oriented interaction framing by borrowing the metaphor of text messaging with autocomplete, and the metaphor of photo messages enhanced by augmented reality, to quickly and accurately allow users to determine the (un)available functionality and invoke it with a single interaction.

1.4 Contributions of This Dissertation

In this dissertation we explore how users discover what is available for them to do in a smart space, and how they can do it, particularly within the currently dominant *proper device name* paradigm. We characterize the fundamental challenges facing user discovery in smart spaces that follow this device-oriented and name-oriented paradigm, including the way that device-centrism limits expectations of interoperability, and the inherently poor guessability of device names. To address these challenges, we design intelligent assistant interfaces that help users in unfamiliar smart spaces discover the assistant's linguistic capabilities and the smart space's functional capabilities simultaneously. While we ultimately achieve our goal of supporting one-shot interactions with intelligent assistants in unfamiliar smart spaces, we

also discuss alternative multi-modal approaches to smart space interfaces that would avoid these challenges altogether and support more natural interactions.

Chapter 2 introduces the literature on discovery, which comes out of psychology. We then review the role of discovery within a human-computer interaction context. We outline various discovery issues that arise in the context of smart homes, voice interfaces, and their intersection. These challenges allow us to formulate a set of requirements for supporting discovery in smart spaces. We discuss how autocomplete and augmented reality can be used as mechanisms to achieve these goals.

In Chapter 3 and Chapter 4, we characterize two key problems with the *proper device name* paradigm. Chapter 3 focuses on the *device-centric* aspect of the paradigm. While there are many ways of presenting a smart home to users, it is not well understood how these different representations impact what end users think a smart space can do, and therefore which representation is the best for supporting discovery. To compare different representations, we formalize them as "interaction framings" that can be used to represent a smart home's capabilities. These interaction framings involve representing smart home capabilities as either a collection of devices or as a collection of datastreams, and then either not personifying the system at all, or presenting an intelligent assistant as a mediating layer on top of those device or data capabilities. We present a hypothetical smart home described using these different interaction framings to over 1,500 participants in an online survey, and ask participants what applications they want in the smart home. We use multiple approaches to analyze the responses, and find that presenting a smart home as a collection of smart devices – a very common interaction framing in modern smart homes – results in limited expectations about automation capabilities. However, layering an intelligent assistant over top of the devices as a mediating layer can help mitigate those effects. These results suggest that focusing on intelligent assistants is the best way to help users perceive the potential value of smart spaces. Consequently, the solutions we explore in the second half of the dissertation focus on interfaces with an intelligent assistant framing. This work was originally published in 2017 in the Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) [63].

Chapter 4 delves into the discovery issues caused by the *proper name* aspect of the proper device name paradigm. Though proper names have often been used to refer to objects in computing systems, we show that they are particularly problematic in the smart space domain. We run a guessability study that illustrates that guessing the proper names of devices is fundamentally hard. Even if devices are assigned the best possible (most easily guessable) names, the odds of a user guessing right in one attempt are still low. More uniquely, the sets of most-guessed names for devices can overlap, so *even if given the list of names* of available smart devices, users will not be able to match a name to its device in one attempt. These results mean we cannot just name devices better—we must design discovery aids.

Chapter 5 and Chapter 6 build up to a solution for supporting discovery in unfamiliar smart spaces that follow a proper device name paradigm. Chapter 5 introduces the "Smart Room Chat Room," where users send text-based chat messages to the intelligent assistant instead of speaking out loud. Using text, a visual medium, allows us to use autocomplete as a

discovery mechanism to help users learn what a room can and cannot do. We outline different design dimensions for autocomplete that could potentially affect discovery, and we compare the effects of several designs. We find that while autocomplete helps people accomplish their goals and brings intelligent assistants up to the same baseline as a smartphone app, there is still room for improvement. We find that even with autocomplete, users have to make trial-and-error attempts to match the names to the devices, and they also cannot determine when a device is *not* smart. The results of this project and the prior guessability study together indicate that simply providing users with a list of the available names is not sufficient, as the names need to additionally be grounded to the actual devices they refer to.

Chapter 6 introduces ARticulate, an interface inspired by ephemeral photo messaging apps that emphasize the user's immediate context. With ARticulate, we finally achieve the goal of one-shot interactions with intelligent assistants in unfamiliar smart spaces. ARticulate works by using augmented reality to show users which devices in the room are smart and what their names are, and to scope autocomplete suggestions while composing messages to the intelligent assistant. In a sense, ARticulate captures the user's gaze while they explore a new space and communicate with the assistant. In addition to achieving one-shot interactions with ARticulate, we also show that ARticulate users are able to successfully use voice-only interfaces afterwards, making it an enabler of the broader interface ecosystem. This ability to teach users transferable knowledge can support users in using other interfaces that lack discovery mechanisms. This work is currently undergoing a cycle of major revisions at IMWUT.

While ARticulate supports discovery under the proper device name paradigm, it also paves the way for interfaces that combine physical context, gesture, and language together—where no single modality contains all of the information, but together the system can disambiguate the user's intent. In Chapter 7, we explore a future where more interfaces incorporate physical context into dereferencing entities in language, and potentially do not even rely on proper names of entities at all. This dissertation concludes that while we should focus on intelligent assistants as smart space interfaces over the long term, language alone is insufficient for successful one-shot interactions in unfamiliar spaces—we should therefore embrace multi-modal smart space interactions as the default paradigm.

Chapter 2

Background and Related Work

The work in this dissertation is informed by prior work on discovery, mental models, proper names in computing systems, and smart space interfaces. In this chapter, we review the literature and synthesize the concepts into a set of requirements for addressing discoverability in smart spaces. At the end, we identify discovery mechanisms from other domains that could potentially be used in smart spaces to satisfy these requirements.

2.1 What Is Discovery?

Psychologists have described discovery as a critical process in human learning. In the 1961 essay "The Act of Discovery," Bruner describes the discovery process in detail, contextualized by his research on childhood development and education [64]. He states that "discovery [...] is in its essence a matter of rearranging or transforming evidence in such a way that one is enabled to go beyond the evidence so reassembled to additional new insights." In his definition, discovery is not merely the random act of stumbling over new facts in the world, but about strategies for search, informed by hypotheses based on the information already known, that direct the learner to the areas of exploration most likely to be fruitful.

In Bruner's view, a critical component of successful discovery is the ability to organize, summarize, and connect information so that the learner can persist in their learning task much longer, rather than being quickly overwhelmed by a barrage of disconnected facts. The ability to synthesize information is also crucial to forming long-term strategies and hypotheses that quickly narrow the search space.

However, a key prerequisite for discovery is suspecting that there is something to discover in the first place: "For the person to search out and find regularities and relationships in [their] environment, [they] must be armed with an expectancy that there will be something to find and, once aroused by expectancy, [they] must devise ways of searching and finding."

2.1.1 Discovery in human-computer interaction

In the context of user interfaces, as outlined by Don Norman, discovery is about the user finding out what actions there are to take and how to take them, while discoverability describes how well interfaces support such user discovery [62].

Discovery interfaces should be interactive, helping the user while they achieve a goal, rather than requiring users to first go through a separate learning phase. In the 1987 paper "Paradox of the Active User," the authors note that empirical studies have shown that many people dislike reading manuals or tutorials without being allowed to act, and will commonly avoid reading altogether whenever possible [65]. Driven by a compulsion for throughput, users, they found, would often rather "jump in" and learn by doing. Paradoxically, the drive to get things done is so strong that such users will preferentially do tasks rather than spend time learning about the system, even if that means potentially doing the tasks less efficiently. Similarly, in trying to teach users how to use voice commands to operate a mobile phone GUI, Corbett and Weber found that a separate tutorial was less effective than providing contextual cues to help users learn "as they go" [66]. This echoes the findings made by Koshy et al., who state that since passenger users are not interested in taking active steps to learn about the smart space, designers should use a "just-in-time" approach to help users discover available actions relevant to their current context [22]. Interfaces should therefore incorporate the discovery process directly into use of the system, rather than treating it separately.

Another concept in human-computer interaction that emerged from the psychology literature on discovery is "sensemaking." Inspired by Bruner and Piaget, Brenda Dervin's work in communications and library sciences in the 1980s introduced sensemaking as a process by which people close information gaps [67]. Sensemaking was brought into the field of human-computer interaction in the 1990s, when researchers at PARC used the term to describe an iterative process whereby a user fits new data into a mental representation, while also changing the representation itself to make better sense of the data or to solve problems more efficiently, resulting in a closely coupled learning loop [68].

Indeed, before the user begins the discovery process, how do they even conceptualize the search space? To begin formulating hypotheses and making sense of experiment outcomes, the user must have some underlying "mental model" of the system.

2.2 How End-user Mental Models Affect Discovery

In "The Design of Everyday Things," Norman defines mental models as conceptual models that users draw from to explain and predict the way that devices will behave in different scenarios [62]. People can use multiple models, and even potentially conflicting models, to explain various aspects of a system. These mental models are influenced not just by exposure to the system itself, but also previous interactions with analogous systems, marketing material, discussions with others, and so on.

Different presentations of the same system might result in different mental models that

differ in terms of which objects are the primary conceptual entities. These differences can persist and affect how users update their understanding of the system even after many interactions. In one study comparing instructions for a laser system, users given one set of instructions thought of the system in terms of switches, and users given a different set of instructions thought of it in terms of components [69]. The two groups' problem-solving strategies centered around either switches or components, even after repeated use.

In psychology, this effect is called *priming*. Priming happens when exposure to an initial stimulus affects the way that a person processes a subsequent stimulus. Some psychologists have used this priming effect to influence the mental models that individuals generate when making decisions about risky processes, from indoor radon to the occupational use of hazardous chemicals [70–72].

One mechanism by which priming could affect mental models is presented by the yoked state space hypothesis [73]. In the yoked state space hypothesis, mental models consist of three parts: 1) the objects in the system, how they interrelate, and what operations can be done on them, 2) the goal space in terms of the user's goals, and 3) a semantic mapping between the goal space and the object space. By changing which objects are the main entities, priming changes the object space and semantic mapping of the mental model, and therefore how users may conduct discovery through that yoked state space to learn how to achieve their goals via manipulations of the system.

The implication of this is that through priming, system designers can influence (either intentionally or accidentally) users' mental models of smart home processes, and thus the discovery process. The ways that different system presentations prime users and affect discovery should therefore be understood and harnessed.

2.2.1 Smart home mental model studies inadvertently prime users with a device-centric interaction framing

Though prior work has examined mental models specifically for smart home users, it has often overlooked the effects of priming, instead treating users' mental models as fixed entities that must be discovered for the purposes of designing intuitive interfaces. However, it is likely that these studies inadvertently primed their subjects to think a particular way about the system.

A number of HCI studies have tried to understand the mental models of smart home users, particularly to support end-user programming in the home. For example, Dey et al. performed a study on mental models so that the authors could build a programming tool to support the way users thought about context-aware applications [74]. However, the study presented end users with a scenario that used a distinctly device-oriented framing to describe the system. The prompt described a generic smart home as a house with "sensors" that could sense the environment and user activities and "execute services on behalf of users." The responses contained an unexpectedly high number of references to objects, surprising the authors—but these results become much less surprising when seen through the lens of

priming. Additionally, the study found that some users perceived the home as a tool while others viewed it as an assistant. This hinted at the possibility that the interpretation of the phrase "execute services on behalf of users" could potentially result in different mental models, but a followup study to explore the effects of different prompts was not performed.

Similarly, another study examined whether trigger-action programming "captures smart home behaviors that users actually desire" [75]. The study presented 318 Mechanical Turk workers with a hypothetical scenario where they had "a home with devices that are Internet-connected and can therefore be given instructions on how to behave," suggesting a strongly device-oriented abstraction of the system, and asked participants for five things they would want their home to do. Results showed that the majority of responses were programs that could be expressed with trigger-action, and most of the remaining responses were remote control interactions. However, the study also found that priming respondents with trigger-action examples had a significant impact on the proportion of trigger-action responses they got back, hinting that priming can have a major effect on the way people think about interacting with systems. While the study explored the effect of priming on the way users *express* programs, it did not reflect upon how the presentation of the smart home in the initial prompt might similarly affect the kinds of operations that users might attempt in the first place. One notable observation is that the responses did not feature any queries. In this work, we find that a lack of queries is characteristic of responses to device-based interaction framings, but queries appear in force when using a data-based framing of the same system. This casts doubt on the assumption that a single-framing study could capture the entirety of behaviors that users desire in a smart home.

To some degree it could be said that the previous studies got out the mental models that they put in. However, the CAMP magnetic poetry paper took a different approach [44]. Like earlier work, the CAMP study assumed that users held "natural conceptualizations of ubicomp technologies," but differed in that the researchers were explicitly aware of the potential to bias users' mental models with the study scenario. To avoid priming users, the study provided comics with pictures and dialog and asked users to describe what they thought was happening in scenes where the parents were "programming" or interacting with the system. The comics were still somewhat biasing because they showed devices and they showed the parents at the computer while programming. However, more concerning is the fundamental assumption that biasing should be reduced as much as possible. The real system *will* be priming, and the system will need to support users' primed mental models. We therefore need to understand how different systems prime users.

2.2.2 A comparative approach to smart space mental models

Instead of trying to avoid priming in mental model studies, it is critical to determine the effects of different kinds of priming so that in designing the system we can select the representations that will prime in ways that we can predict and support. However, so far there has been a dearth of comparative studies that embrace priming as part of the design process.

In Chapter 3, we formalize a notion of "interaction framings" for smart homes. This allows us to compare different ways of representing a smart home's capabilities to users. We categorize our framings along two independent axes: 1) capabilities, where the system's functionality is presented as either devices or datastreams, and 2) personification, where the system is either mediated by an intelligent assistant or unmediated. We compare four interaction framings that represent the extremes of each of these choices. In our comparative study, we find assistants promising for the long-term future of IoT. This grounds and contextualises our approach in future chapters for improving smart space dicoverability.

2.3 Proper Names in Computing Systems

Smart homes are far from the first computing system built around proper names. Command lines, databases, and file systems all use names to specify actions or entities.

Computing systems that rely on proper names to reference data objects have long struggled with discoverability. In the 1980s, Furnas studied what he called the vocabulary problem in computing systems, and found that across many domains, the guessability of even the best proper name was very low [76]. He also identified what he called "the precision problem," wherein a the same proper name could plausibly refer to multiple data objects. The more guessable the name, the more generic it is and the more objects it could potentially refer to; the more specific the name, the less guessable it is during discovery.

2.3.1 Past solutions for proper name discoverability

Different computing systems have developed different solutions to the vocabulary problem, but most involve some kind of visual aid or pointing-based selection.

Autocomplete, which provides real-time visual feedback to users as they type, is one such solution. Autocomplete has been used to help users discover available proper names of tables, fields, and operations in databases [77], and proper names of actions and objects in command lines [78, 79]. These days, popular operating systems such as Windows and MacOS provide a search bar augmented with autocomplete suggestions for finding files by name. We discuss autocomplete further in Section 2.6.2.

Upon the introduction of graphical user interfaces, the notion of a cursor allowed users to select an object by pointing to it, rather than having to specify a proper name. These days users can launch applications by clicking icons rather than typing names on the command line, and they can find and select files by clicking through directory hierarchies in a file explorer. These objects are usually also labeled with their proper names, but users only need to recognize the name of a desired object, rather than recall it. This approach of physically interacting with iconic representations of digital objects is called "direct manipulation" [80–82].

Voice interfaces have been combined with cursors to overcome the need for proper names. MIT researchers allowed users to point to a large projector screen to place a cursor while using voice commands to operate the visuals on the screen. The presence of the cursor on the

screen allowed users to use pronouns in their commands rather than having to specify proper names [83]. The cursor concept has been explored in ubiquitous computing environments, with projects such as WorldCursor and WorldGaze [84, 85]. Our use of augmented reality to display smart devices to users and encourage users to point their phone at smart devices they want to interact with follows in this tradition of gesture-based selection of interactive digital objects.

2.4 Smart Space Discoverability

While there has been a lot of work on designing usable smart home interfaces in situations where users are familiar with the underlying capabilities, less work has been done to study the scenario where people walk into an *unfamiliar* smart space and must discover what capabilities are available and how to use them.

In "Designing for Serendipity," Newman et al. identify discovery as a major problem for recombinant computing [86]. Recombinant computing is a vision of ubiquitous computation where end users opportunistically discover nearby services and connect them to perform tasks. Fundamental to the recombinant computing paradigm is the assumption that available devices and services are likely to change, and they may be used for multiple different purposes. Consequently, the authors highlight the need to provide users with information about what is currently available and what it can do. Their recombinant computing system, SpeakEasy, exposes human-readable information about what devices and services are available, their capabilities, and information about their relationships to the environment, such as location. This information is presented in a browser using drop-down menus that can be organized along several attributes, such as device type, location, or owner. Hints about how these components can be combined to perform tasks is provided in the form of templates. In Chapter 6, our autocomplete mechanism similarly organizes capabilities in a dropdown menu style, grouped into categories intended to aid users in quickly forming an overall picture of what is available.

In the paper "Let There be Light," Brumitt and Cadiz compare potential smart space interface modalities to determine which would be best for intelligent home environments [87]. They look at five different interfaces for controlling lights: a text-centric touchscreen interface that lists the names of lights grouped by location and provides sliders for dimming; a graphical touchscreen interface that shows an unlabeled tappable icon for each light placed on a floorplan of the room; a voice-only interface; a voice interface with location awareness; and a voice interface with location and gesture awareness. To free the voice interfaces from technological constraints, they were implemented using a Wizard of Oz method where a hidden researcher interpreted the user's utterances and performed the desired actions.

The researchers were interested in seeing how people would try to interact with the system upon first entering the room. Participants were assigned a task to set up the room for a friend's surprise party and told only that the environment is intelligent. Given this open-ended prompt, only one of their 18 users initially tried to use speech, while the others walked around

to find a device that seemed like it could control the room. After this first task, participants were asked to do the same task with each interface. At the end, participants were asked to do the task one last time using whichever interface they preferred. Afterwards, participants were asked to rate the interfaces on how much they liked it and the perceived ease of use.

The interface that closest resembles today's ubiquitous device-oriented smart phone apps, the touchscreen interface that displayed light names and sliders, was rated the least-liked and least easy to use. As the researchers observed, "one major issue with any such display is creating consistent, clear labels to describe individual lights."

The voice interface was rated the most-liked, the easiest to use, and was by far the most popular choice for the final task. However, the researchers caution that "people liked speech because it worked nearly perfectly, and it worked nearly perfectly because we had a 'wizard' (another researcher) controlling the lights to make sure that they responded as perfectly as possible to user requests."

The authors predict that the voice interface would be difficult to implement in practice due to the widely variable language that users used to refer to devices. Users commonly used indirect references ("turn on the light over there in the corner" or "turn on this light"), references relative to another object ("turn on the light above the couch"), or relative direction with an implied frame of reference ("turn on the left light"). This is a type of language use is called *deixis*, where the contextual information for resolving the entity reference exists outside of the text of the utterance itself. Over 57% of the speech commands that participants used involved some kind of deictic phrase to refer to the lights. A voice system that only has access to the text of the user's speech would struggle to select the correct light.

The researchers reject the proper device name paradigm as a good solution to this problem. They point out that labeling all the lights with proper names would "[require] a person to learn a naming scheme for all the lights in the house, and visitors will be faced with the problem of trying to guess the labels selected by the owner."

Instead, they propose a multi-modal method to resolve the deictic object references – in 91% of the tasks, the participants looked at the device they were referring to. This means that gaze direction can be used to resolve most ambiguous device references, without the users having to know any proper device names. In Chapter 6 we incorporate this into our augmented reality approach, which can be seen as a way of capturing user gaze.

Unfortunately, modern smart homes have coalesced around the proper device name approach, likely due to the ease of implementation. The "Let There Be Light" study did not evaluate how the usability of voice interfaces would be impacted if the system only understood proper names, like current systems. In Chapter 4 we characterize how fundamentally difficult it is to guess proper names in smart spaces—a major usability obstacle in an unfamiliar environment.

2.5 Discoverability Requirements for One-Shot Interactions in Unfamiliar Smart Spaces

Or goal is for users to walk into an unfamiliar smart space and accomplish a smart space task in a single attempt, or, if the task is not possible, not attempt to do it at all—what we call a "one-shot interaction." Additionally, while performing a specific task users should learn broader information about the room that can help with other potential future tasks that may be undertaken with alternative interfaces that lack discovery aids.

To have a successful one-shot interaction and learn about the overall smart space, users must acquire certain pieces of information about the system. We categorize the information users must learn into three types: semantic, lexical, and grounding information. Interfaces that support discovery in smart spaces should present this information to the user as they accomplish desired tasks, and the information should be transferable to other interfaces.

2.5.1 Semantic discovery

We use the term *semantic discovery* to refer to discovering the valid states of the system. In smart spaces, users must be able to answer two questions. The first question is, "Is this device interactive or not?" In most smart spaces, some devices may be smart and connected, while others are regular appliances that cannot be controlled by smart space interfaces. Users in an unfamiliar smart space need to be able to quickly form a mental map of which devices are digitally connected and which are not. The second question is, "What are the states this interactive device can be in?" For example, some smart light bulbs can only turn on or off, some can change shades of white, while others can change colors and display animations. For users to formulate goals, they must learn what is possible.

2.5.2 Lexical discovery

In smart spaces designed around a proper device name paradigm, figuring out how to put a device into various desired states means needing to learn the proper names, especially of the device, but also sometimes the state (e.g., "periwinkle blue" is the proper name of specific color setting). We call the discovery of the system's proper names *lexical discovery.*

2.5.3 Grounding discovery

Knowing the possible states of the devices and knowing the special keywords of the system is not enough. The user also needs to know which names belong to which devices or states. Is "Reading Lamp" the light on the right of the couch, or the left? What actual color does "periwinkle blue" refer to? We refer to mapping names to their objects as *grounding discovery.*

2.5.4 Just-in-time presentation

As we discussed in Section 2.1.1, discovery should happen while the user is using the interface to do tasks, rather than in a separate learning-only context. There is an inherent tension between discovery (exploration) and doing (exploitation). Nevertheless, we think we can balance these two goals by designing an interface that helps users accomplish targeted tasks, but in such a structured way that they end up learning about the overall space, almost as a unintentional side-effect. The question is whether such designs can also be fast and frictionless.

2.5.5 Transfer of knowledge to other interfaces

There is a diversity of smart home interfaces available to users, and different interfaces are useful for different things at different times [87]. Users should have the option of being able to use whichever interface most suits their needs in the moment. This means that any discovery mechanism must also help users learn broadly, so that they can transfer the knowledge to another interface that does not have autocomplete. After using the discovery-supporting interface for a few tasks, they should have enough understanding of the space's capabilities to successfully perform new tasks with a different interface, even if it does not support discovery. For example, if you use a discovery-oriented phone app once or twice, you should later be able to think of a new task (knowing that the room supports it) and use a voice interface to accomplish the task, despite the lack of discovery aid.

This is significant. We do not need to force smart space occupants to use a discovery-oriented interface all the time, just during a discovery phase that then enables them to switch to something else. Any interface we propose is not intended to be the categorically best smart home interface, but rather an addition to an ecosystem that all works together to support the user. A discovery-oriented interface would unlock the rest of that ecosystem by supporting learning that transfers to the others.

2.6 Potential Discovery Mechanisms

The solutions we explore for one-shot interactions will be centered around interactions with intelligent assistants, based on the results of our study in Chapter 3. To best support discovery, we forego a voice approach to focus on a visual approach. This allows us to use discovery mechanisms like autocomplete and augmented reality.

2.6.1 Why voice is insufficient

If we are targeting intelligent assistants, which people primarily communicate with using voice input, why not have an entirely voice-based discovery solution?

Many have noted that discovery is a significant obstacle to voice interface usability [76, 88, 89]. Voice-only discovery solutions have focused on prompts that expand or condense options

and instructions at appropriate times [89–91]. However, prompt-based approaches tend to take worst-case time for initial interactions, imposing a potentially prohibitive overhead cost to interacting with each new space. Speaking on the usability of speech interfaces [92], Fang Chen observes:

> "If the user is still discovering the functional capacities of the underlying software then a verbal prompt sequence becomes very unsatisfactory. Such a presentation technique leaves no room for review, comparison and thought. Exploration becomes impractical. Thus the first human factors issue that needs to be addressed is to devise an approach that allows dialog with users that has the structural advantages of menu interaction and the directness of free form spoken natural language."

Consequently, past work has recommended mixed voice-visual interfaces due to the ability to convey much more information through the high-bandwidth visual channel [28]. M-VUI was one such mixed-modality system that combined voice with the direct manipulation of a mobile interface [66]. To aid discovery, users primarily used the direct manipulation interface, which showed users the equivalent language-based command for any task they performed. If the user wished to learn more, M-VUI also exposed contextual help menus that showed users all available options when requested. Though we also use a graphical interface to help users learn linguistic information, we forgo a direct manipulation approach in favor of maintaining the consistency of an intelligent assistant interaction framing.

There is an established visual way to communicate with intelligent assistants: messaging. The three major intelligent assistants provide text interfaces in addition to voice. Having users send messages to the assistant using a mobile device preserves the central interaction framing, while allowing us to use visual discovery mechanisms to aid in message composition, including autocomplete and augmented reality.

It must be acknowledged that messaging is not equivalent to voice. It lacks the advantage of hands-free operation, albeit while gaining the advantage of quiet execution. However, our goal is not to replace voice, but rather to complement it, and even enable its use. We envision users using the messaging interface for performing smart space tasks with an intelligent assistant in an unfamiliar space, which is a situation in which they could not successfully use voice anyway. However, once they have learned about the space through initial use, that knowledge should allow them to use voice or device-centric apps or whatever else they prefer. In evaluating our solutions, we examine this transfer of knowledge, and find that our messaging solution can enable successful use of a voice interface that was previously difficult, if not impossible, to use.

2.6.2 Autocomplete for discovery

Autocomplete is a list of possible valid user input text that updates dynamically based on what the user has typed, and which is visually co-located with the input widget. While

autocomplete has become ubiquitous in modern life, there are few studies examining how autocomplete design impacts users [93, 94].

In daily life, autocomplete is primarily used not to aid discovery, but rather to provide predictions of what the user is already planning to type in order to reduce typing time and effort. One of the earliest predictive autocomplete interfaces was the Reactive Keyboard, developed in the 1980s to reduce the number of physical gestures needed from users with mobility impairments [95, 96]. However, autocomplete became mainstream when Google introduced it for Search in 2004 [94]. Google explicitly states, "we call these autocomplete 'predictions' rather than 'suggestions,' and there's a good reason for that. Autocomplete is designed to help people complete a search they were intending to do, not to suggest new types of searches to be performed" [97]. Similar approaches have been used to provide predictions in many other domains, such as browser URLs, Unix commands, and email [78, 98].

However, the predictive approach only makes sense for applications where users already know what they want to type and do not care about the rest of the dataset (sometimes called "known-term" tasks [99]). In an unfamiliar smart space, we do not want to predict what the user wants to say, as our assumption is that the user is not yet sure what they *can* say. To help users interact with an unfamiliar smart space, our goal is to teach users about the *overall* set of things they can say, so that they can learn what is and is not possible, and also recall the capabilities and names later without the discovery interface in front of them.

Google Assistant [100], which powers Google Home [101], illustrates how prediction-oriented autocomplete does not serve smart space discovery in practice. While users can speak to Google Assistant through the phone or a smart speaker, users can also *type* Home commands through the app, with the interface providing autocomplete suggestions. While this seems similar to our proposal on its surface, the suggestions are drawn from population-wide statistics and the user's typing history, and suggest common words or phrases, rather than helping the user discover commands for devices that are actually in the room. Additionally, it exposes only the most likely words instead of providing insight into all the options so that users can learn more broadly. These prediction-oriented features do not help new users explore an unfamiliar smart space.

Though less common, autocomplete has also been used to support more exploratory workflows, where the technique is sometimes called "autosuggest" [99]. Autosuggest features help the user to discover similar or related queries.

One early example of autosuggest emerged in information retrieval, particularly in real-time interactive query expansion (RTQE). Query expansion helps users refine keyword queries to a document store in order to improve precision and recall of retrieved documents [102, 103]. In RTQE, suggested keywords are shown alongside the user's query as the user types [99].

Autocomplete has also been used to help users explore database schemas [77]. Real-time feedback while constructing database queries help users discover the structure of the underlying database (the schema) at the same time as providing suggestions for the specific query itself. This is very similar to the smart space discovery task, though in our case, users are constructing natural language sentences rather than structured queries.

Code autocompletion in integrated development environments (IDEs) also shares simi-

larities with the smart space domain. In both cases, the set of autocomplete possibilities at any moment is relatively small and constrained by the syntax and types of the associated text. Autocomplete suggestions for variable names are specific to the unique variable names available in the project, rather than chosen from population-level statistics. Suggestions for the methods available to an object are usually comprehensive and alphabetically ordered, allowing developers to scroll through the complete list to discover API functionality [104]. While code autocompletion is among the most commonly invoked IDE features [105], how well it helps developers discover and retain API knowledge has not been evaluated.

A major limiting factor of these prior uses of autosuggest for smart space discovery is that these autocomplete interfaces assume that the user will always have access to them, so the suggestions are often still geared towards supporting the user's specific goal, rather than preparing the user to accomplish unrelated tasks in the future without the autocomplete interface. The novelty of our autocomplete approach lies in designing autocomplete suggestions to provide as comprehensive a picture of the space as quickly as possible to allow the user to switch to other smart space interfaces, such as voice, that may be more contextually convenient (e.g. fast, hands-free) but which may not have discovery mechanisms. In our studies, we ask users to accomplish specific smart space tasks using various autocomplete designs, and then test the users afterwards to see what information they can recall, including other information about the room that they did not use directly in their task.

2.6.3 Augmented reality for discovery

Augmented reality, first proposed by Ivan Sutherland in 1968, is a way of visually embedding virtual objects in the physical world [106], including informative readouts about physical objects [107–110]. Augmented reality has been used for many applications, such as military, advertisement, industrial assembly, maintenance and inspections, 3D design, gaming, medicine,sports, education, office collaboration, and emergency response [111–114]. Thanks to improvements in mobile technology, particularly in computer vision, augmented reality has recently become widely supported by major smartphone platforms [113, 115]. This makes augmented reality a promising avenue as a mechanism for discovering capabilities available in a smart space and what their names are.

2.7 Camera-based smart space interactions

To incorporate both autocomplete and augmented reality into an intelligent assistant interaction, we use a messaging metaphor that draws on users' prior experiences with existing messaging workflows. We model our solution, ARticulate, on ephemeral photo messaging apps, in which communication between users centers around photographs of the immediate physical context [116–119]. Within this kind of photo-oriented messaging workflow, we can use augmented reality to provide visual feedback in the live camera view that indicates to users whether a device is smart, and if so, what its name is. Autocomplete is employed to

reveal the functional capabilities of each device while the user composes the photo caption. The final message can be sent to the intelligent assistant. The combination of augmented reality and autocomplete provide an interactive means for users to learn semantic, lexical, and grounding information that can potentially transfer to other "proper device name" interfaces, thus satisfying all of our requirements for smart space discovery.

One smart space interface that also uses photo-taking as a way of getting a handle on smart devices is Snap-to-it [120]. Snap-to-it is an intuitive way of accessing a user interface for a smart device by snapping a picture of it. The system uses computer vision to identify which of the known smart devices is in the photograph, and then provides the user with an interface for that device. While the core interaction is appealing, the design of Snap-to-it does not support discovery in an unfamiliar smart space at all. Users have to know (or at least suspect) that a device is interactive in order to take a picture of it, but the system does not provide any hints. To aid discovery during the study, the researchers posted fliers stating that there was a smart device in the area. Snap-to-it also does not teach users information, such as proper names, that would enable the use of other interfaces. Finally, Snap-to-it lacks an intelligent assistant interaction framing. Our approach addresses all of these shortcomings, helping users autonomously discover and use capabilities in an unfamiliar smart space.

WorldGaze, which is contemporaneous with our work, introduced a method for using a smart phone to determine what a user is looking at while speaking with a voice assistant [85]. WorldGaze determines the orientation of the user's head from the front-facing camera and projects a vector into the rear camera's view. When using WorldGaze to interact with smart devices, the user holds up the smartphone, which shows the rear camera's view with the gaze direction indicated by a red line projecting into the world. Smart devices in the camera view are labeled using augmented reality. While holding up the phone and pointing their head at the target device, users speak aloud to an assistant. The gaze information allows users to use pronouns or omit names entirely while issuing voice commands.

While WorldGaze seems to address the same problem as ARticulate, there are a number of differences. Most significantly, WorldGaze does not expose the functionality of the devices, such as whether the lights support dimming or different colors, and what phrases the assistant is guaranteed to understand. ARticulate does this using autocomplete. As currently implemented, WorldGaze also cannot support combining gaze with text, because the user cannot type while pointing the phone and looking. ARticulate does this by allowing users to quickly snap a photo and then lower their arms to type. As we show later, text has advantages over voice in certain contexts. Finally, WorldGaze does not prioritize teaching knowledge that will transfer to other interfaces. The augmented reality annotations did not provide the proper names of the devices, but rather the general class of device, such as "light" or "speaker." In a future where every smart space interface supports name-free interaction this may suffice, but while we live in a world where some interfaces are built around proper device names, there is an advantage to teaching users the proper names. In addition to supporting seamless interactions with the assistant, our solution helps users learn an overall mental map of the device capabilities and device names in the room, which can potentially transfer to other interfaces.

2.8 Summary

Discovery is the process by which people structure and acquire information in order to learn something new. In human-computer interaction, user discovery involves formulating mental models of a system to represent information about how it functions, and within that model, discovering what actions are available and how to invoke them.

Modern smart spaces can be represented using a number of different conceptual models, so in the next chapter we explore the effects of these different interaction framings on mental model formation and end up taking an intelligent assistant approach in our discovery solutions.

In modern smart spaces designed around a proper device name paradigm, discovering what actions are available and how to invoke them means learning which devices are smart and what their proper names are. To help users discover that information while communicating with an intelligent assistant, we will use messaging as the medium instead of voice, which will allow us to use visual feedback such as autocomplete and augmented reality as discovery mechanisms.

Our ultimate goal is a smart space interface that allows people to interactively discover the overall capabilities of an unfamiliar smart space while also accomplishing specific tasks in that space. Since smart spaces are multi-modal with many interfaces that each have different advantages, the solution must also strengthen the ecosystem. Use of the interface should reveal information that enables the later use of other interfaces that may not have their own discovery mechanisms. By providing information about the overall capabilities of the space, the proper names of the devices, and which names correspond to which device, we expect that users will not only be able to perform one-shot interactions in unfamiliar smart spaces, but also use other interfaces successfully afterwards.

Chapter 3

How Device-Centrism Limits User Expectations of Smart Space Capabilities

Content in this chapter originally appeared as Clark, et al. "Devices and data and agents, oh my: How smart home abstractions prime end-user mental models." IMWUT (2017).

While smart space architecture may take many forms, every potential solution must abstract the system to provide interfaces for end-user interaction. The interaction framings we use to present the system collectively reflect some conceptual model or metaphor for how the user is expected to interact with the smart space. For example, a smart home app that abstracts a smart lighting system by providing virtualized interactive representations of the individual bulbs in the app conveys a device-oriented model of interactions with the system, whereas an app that exposes the lighting system's state as a datastream that emits and receives messages provides a data-oriented model. These interaction framings can be explicitly chosen by designers to guide the users in interacting with and understanding the system, or they can reflect implicit assumptions on the part of the designers about how the system works.

There are many different smart home interaction framings for smart space system designers to choose, but little work has been done on understanding the impact of different system interaction framings on end users' mental models and expectations of these systems. Previous attempts to better support end-user interactions have attempted to "get at" end users' mental models for smart homes as though these models are an independently existing property of users, without considering the effects of priming.

To develop interfaces that support end users, instead we need an intentional and conscious comparison between several different system representations. To that end we propose a framework that would allow comparisons between different conceptual models, which breaks conceptual models down into several independent dimensions whose possible values we call *interaction framings*. This framework allows us to make comparisons between conceptual models and understand how the different ways we can present the same underlying system

will affect users' expectations and behavior.

In fact, there is evidence to suggest that the interaction framings that we are using are preventing consumers from thinking of valuable integrated applications that utilize the full range of a smart spaces's capabilities. According to Affinova's 2014 consumer report on IoT adoption, users think the current smart devices on the market are "gimmicky" [121]. While some devices may be gimmicky single-purpose products, many others can be integrated into applications by various emerging platforms. It is possible that due to the way these systems are presented, users do not even think certain useful applications are possible or within the scope of the system.

Additionally, the interaction framings we use can create a gap between expectations and reality that prevent adoption. An understanding of what expectations different interaction framings encourage can allow us to manipulate and shrink this gap, both on the user side and on the system side.

In this chapter we make several contributions. First, we demonstrate that the interaction framings chosen to represent a smart home system have significant priming effects on end users. We created a set of questionnaires depicting the same hypothetical smart home in four different ways and deployed the questionnaires via Mechanical Turk. Each worker was presented with only one of the four descriptions, and then was asked to describe the applications they wanted. We collected responses from over 1,500 participants. Though mental models cannot be directly observed, they can be thought to emit signals via various user behaviors. To gain insight into the mental models at play, we analyzed the following signals: 1) the qualitative differences in responses, 2) the characteristic words for each set of responses as determined by χ^2 feature selection, and 3) the differences in the *operation profile* of the responses, which we define as the distribution of tasks that a group of users performs on the hypothetical system – for example, the relative proportion of immediate actions, questions, conditional actions, and notifications. We find that users' mental models and the resulting operations that they attempt are heavily affected by the interaction framings used to present the system. This finding highlights how critical it is to consciously choose interaction framings, both during system design and when creating scenarios to study users' mental models for a particular domain.

Second, we describe what the specific priming effects (and therefore design trade-offs) are for some common interaction framings. We examine the effects of two different dimensions for interaction framings: How the system's capabilities are represented ("devices" vs. "data"), and whether or not the system is personified ("assistant-mediated" vs. "unmediated"). Combining these two framing dimensions results in four distinct conceptual models: Unmediated Devices, Unmediated Data, Assistant-mediated Devices, and Assistant-mediated Data.

While these four archetypal interaction framings are far from the only possible system interfaces, and in practice are not mutually exclusive, they are a good starting point for understanding the effects of commonly-used interaction framings on end-user thinking. Understanding the different workloads and expectations engendered by common interaction framings for smart spaces will help system designers better understand the trade-offs involved when choosing the interaction framings for their system.

Finally, we also released our complete dataset and code to the community for additional analyses, with IRB approval. An archival link is provided in Appendix A.

As far as we know, this is the first substantial public corpus of end-user descriptions of desirable IoT applications, as well as natural language commands and queries directed to smart home assistants. Researchers can use the data to improve the relationship between the smart spaces and their users, for example by building better natural language interfaces, creating desirable applications, and gaining additional insights into end-user needs and concerns in the smart home domain.

3.1 A Framework for Mental Models and User Interactions

In this section we review the relationship between conceptual models, mental models, and user operations. By understanding how these elements interact with each other, we can lay the foundations for a methodological approach that will allow us to compare the impact of conceptual models on end users' expectations and behavior.

As described by Norman, designers have some implicit or explicit notion for how users should think about system, particularly with regards to what the different parts are and how they work together. This notion is called the *conceptual model*, or sometimes the *interaction metaphor*. The designer conveys the conceptual model to the end user through the *system image*. The system image is everything about the system that is visible to the user, which consists of two main parts: 1) the concrete products and interfaces created by the designer, which collectively convey the designer's conceptual model or interaction metaphor, and 2) external factors that the designer does not control. The first part includes artifacts like the physical system interface and documentation, and the second part includes the user's prior experiences, news reports, anecdotes from friends, and so on.

As users perceive the system image, they generate a mental model for how their interactions affect the system and how the system affects them. Prior work has decribed mental models as a "yoked state space" where users form a mapping between a device (or system) state space to a goal state space [122]. The conceptual model, as a manifestated by the system image, is therefore critical in influencing what states the user perceives the system to have and what goals the user thinks the system can satisfy.

While a user's mental model cannot be directly perceived, the types of operations that the user assumes are available and would like to employ can give us some insights into their mental model. In the yoked state space theory, users perform operations on a device in order to navigate the device state space to reach states that, according to their model, map to desired goal states [122]. This means that by observing the kinds of operations users expect to be able to perform, we can get a sense for how they are internally modeling the system states and what goals they perceive the system as being able to satisfy.

Understanding what kinds of operations users will assume are available when provided with

a particular conceptual model would help system designers minimize the "gulf of execution" that separates user goals from the actions they need to take to make the system satisfy those goals. By knowing what operations users will expect to able able to do, system designers can design their system to support those operations as first-order primitives. In the opposite direction, designers can also discover which conceptual models will align user expectations and operations with what the system best supports.

Given the relationship between conceptual models, mental models, and operations, it should be possible to present different conceptual models to users and observe how the types of operations they perform change due to the different mental models they generate from the system image. However, how can we control for those aforementioned external influences that are also a part of the system image, like prior experiences? In this paper we overcome that challenge by looking at population-level distributions of operation types. When comparing what populations as a whole attempt to do in response to a given conceptual model, the individual variations in external influences essentially come out in the wash. A population-level analysis is also useful because smart home system designers often design a single system that will be used by a large number of people, in which case it can be beneficial for designers to know in advance what the system's overall workload will look if it is presented in a certain way.

3.1.1 Interaction framings framework for conceptual models

To understand the trade-offs between different conceptual models, we need to articulate a set of dimensions that we can use to describe and compare them. We propose breaking conceptual models down into several independent dimensions that each have a set of possible values called *interaction framings*. Under this approach, conceptual models can be constructed or described by selecting an framing along each dimension. While many such dimensions may exist, in this paper we focus on two: *capabilities* and *personification*.

Capabilities. Conceptual models can use many different interaction framings to represent a smart home's capabilities. A common approach is to represent capabilities in terms of the available devices, but it is also possible to describe available functionality using other fundamental conceptual entities, such as actions, data streams, facts, or events. In this work, we compare two interaction framings for capabilities, which we call "devices" and "data."

Because smart space systems are frequently conceptualized as interconnected devices or physical artifacts at the architectural level, it is common for system designers and even HCI researchers to assume that the higher layers presented to users will share a similar device-oriented framing. In particular, there is a great deal of IoT literature that assumes device-oriented interfaces will bubble up to users from the lower layers [123–126]. However, there is no reason that the higher level of framing presented to users needs to resemble the ways that system designers think about the system. In fact, in this work we show that depending on the designer's goals and intended audience, using a device-oriented framing may not always be the most beneficial way to model and present the system.

Personification. We also examine two interaction framings that can be used to represent different degrees of system personification: "unmediated" and "assistant-mediated." An unmediated system is one with no personification layer, and an assistant-mediated system is one with an intelligent assistant that acts as an intermediary between the user and the rest of the system. Prior work has shown that users will interact with functionally-equivalent systems differently depending on whether the system is personified as human-like assistant or not. For example, while both Google Now and Siri are phone-based voice interfaces, research has found that users treat Google Now like "voice-activated search," whereas they treat Siri like a social actor due to the latter's presentation as a human-like assistant [127].

There are other dimensions that we do not examine in this work, but which would nevertheless be useful in an interaction framing framework. For example, the dimension of *initiative* is independent from personification. Systems can behave with initiative without being personified as human-like. Prior work has compared the user experience of systems with different degrees of initiative in a variety of domains [128–130]. While these systems act with various levels of agency, they are not presented in an anthropomorphic fashion.

There is also the dimension of *input modality*, which can be visual, written language, voice, gesture, tactile, and more. Some related works have compared the user appeal of different modalities in the smart home domain, such as voice, voice and gesture, touch screen GUIs, and touch screen text interfaces [87]. We do not explicitly compare the effects of these different modalities on user thinking, though we do look at what modalities users assume given interaction framings along other dimensions. We leave exploring the effects of additional dimensions beyond the two we consider to future work.

While conceptual models are used implicitly thoughout smart home research, we are the first to attempt to explicitly characterize and compare them systematically. Framing dimensions such as the ones we propose here could form the basis of an evaluation framework that allows designers to compare the impact of various conceptual models and make claims about why one particular framing would be a better choice for an interface than another.

3.1.2 Classification of user operations

In order to determine how conceptual models affect the kinds of operations that users would like to perform, we must next be able to classify user operations. We separate smart home interactions broadly into *immediate interactions*, which complete their execution right away, and *conditional interactions*, which may result in interactions at a later time.

Immediate interactions. Remote control commands like "turn on the lights" are one form of immediate interaction. However, since there are other immediate requests for action that may not fall under a remote control mindset, such as "wake up my children," we use the more general term *immediate actions* to refer to this subcategory of operations. In addition to actuation there are also queries, an often-overlooked form of smart home interaction. Immediate queries can take two forms: *direct questions* and *indirect questions*. Direct questions are any query that would properly end with a question mark. Indirect questions are requests for information that are formulated as a command, such as "tell me how much I

weigh." We make the distinction between the two forms to highlight that in our operation taxonomy we consider indirect questions to be requests for information rather than actions.

Conditional interactions. Trigger-action statements, best exemplified by "when I come home, turn on the lights," are one type of ongoing interaction. However, since there can be other ways to express conditions on actions besides event-based triggers (using words like "until," "unless," "before," and "while"), we use the more general umbrella term *conditional actions*. Just as it is possible to have conditional actions, it is also possible to have conditional queries, which we call *notifications*. The difference between notifications and indirect questions can sometimes be subtle. A good rule of thumb is that indirect questions are usually requests for facts, and notifications are usually requests to be informed of events. For example, "tell me when my husband gets home" is most likely a notification, whereas "tell me when my husband will get home" is an indirect question.

This is not meant to be a definitive operation taxonomy for the smart home. However, we believe that including both actions and queries is an important step for developing a more comprehensive way to understand mental models. By disinguishing between requests for actuation and requests for information, we hope to be able to tell a more complex story than an analysis of just actions or just queries would allow.

3.2 Evaluating the Effect of Interaction Framings on Mental Models

In this work we look at interaction framings along two dimensions: whether or not the system is personified by an assistant who acts as an intermediary ("unmediated" or "assistant-mediated"), and whether capabilities are represented as available "devices" or as available "data." The possible combinations along these two dimensions result in four different conceptual models: Unmediated Devices, Unmediated Data, Assistant-mediated Devices, and Assistant-mediated Data.

To examine the impact that each of these four conceptual models has on users' mental models, we devised questionnaires describing a hypothetical smart home using each of the conceptual models. We presented Mechanical Turk workers with one of the four questionnaire prompts and asked them to write either about what applications they would want in their smart home or what they would want their assistant to do. We then analyzed the entities and operations present in the written responses.

Given only a few minutes with a questionnaire prompt, users cannot explore a conceptual model to the same depth as they can given a more fully-realized system over a longer period of time. Previous work has shown that there is a smart home app development lifecycle that begins with brainstorming the application, as we ask users to do, but then also includes stages where the user iteratively improves it as it runs in the house and bugs or unexpected behaviors emerge [27]. Our questionnaires do not capture the debugging phase, which means missing certain kinds of interactions like introspective queries (for example, the word "why"

doesn't appear once in the assistant responses, even though in practice users do want to know "why" things happen). Additionally, our methodology cannot rule out the possibility that the priming we see is a minor initial effect that can be overruled with training and experience interacting with the system.

However, prior work has shown that initial mental models formed by users before they even interact with a system can have an ongoing influence over how users update their understanding of the system based on newly-acquired knowledge [69]. These persistent effects, combined with the fact that even our brief descriptions resulted in observable differences, indicates that presentation matters, even if more sustained engagement might shape the users' mental models even further. Indeed, even if the initial impressions do not have a lasting effect on user interactions once the user has become accustomed to the system, these primed mental models still affect purchasing and adoption decisions. A user should be able to look at the presentation of a system and envision valuable interactions before actually interacting with the system.

3.2.1 Questionnaire design

We devised two scenarios to collect data about the way end users describe smart home applications. The unmediated scenario asked respondents to prepare to imagine IoT applications, and then gave one of two treatments with equal likelihood: 1) a list of smart devices that users had at their disposal in their smart home broken down by sensor, actuator, or online service, or 2) a list of data streams that could reasonably be synthesized from the devices listed in the other prompt, broken down by read-only data (sensor readings), read-write data (actuator statuses), or online service data.

To create the two sets of capabilities, we first generated a list of common smart home sensors, actuators, and online services, which became our list of devices. Then we generated the list of data streams by converting each device from the first list. For example, while the device list has "Controllable RGB Lights," the data list has "Whether the lighting is on or off," "What color the lighting is," and "How bright the lighting is." The full list of devices and data streams that we provided can be found in the online appendix.

In order to give respondents time to look over the list of capabilities, they were not allowed to continue until a one-minute timer expired. On the next page they were asked to write for five minutes in a text box about what kinds of applications they wanted in their smart home, with the list of devices or data streams displayed above the text entry box for inspiration. Afterwards, we asked the participants basic demographic questions and assessed their general familiarity with IoT and technology.

The assistant-mediated scenario introduced an artificial intelligence assistant (an "AI") as the intelligence behind the smart home controls. In this scenario, respondents were first asked to select a gender for their AI's voice from a randomly ordered list of male, female, and androgynous, then respondents were asked to name their AI. Our narrative highlighted that the assistant was trustworthy and wanted to help the participant. We then gave respondents one of two prompts with equal likelihood, just as in the unmediated scenario: 1) a list of

Personification	Capabilities	Conceptual Model	Responses
Unmediated	Devices	Unmediated Devices	313
	Data	Unmediated Data	302
Assistant-mediated	Devices	Assistant-mediated Devices	442
	Data	Assistant-mediated Data	478

Table 3.1: Overview of questionnaires. We administered questionnaires with two scenarios on Mechanical Turk, each of which had two possible treatments, which resulted in four unique prompts describing smart home conceptual models. The unmediated scenario asked what applications end users wanted in their hypothetical smart home, while the assistant-mediated scenario asked end users to tell a hypothetical smart home AI what they wanted it to do. For each scenario, the smart home's capabilities were described either by a list of devices or by a list of data streams. Participants were only presented with one of the four conceptual models.

devices that the assistant had access to in their smart home broken down by sensor, actuator, or online service, or 2) a list of data streams that could reasonably be synthesized from the devices listed in the other prompt, broken down by read-only data (sensor readings), read-write data (actuator statuses), or online service data. The respondents were given a minute to look over the list. Afterwards, we told respondents that their AI wanted to help them by automating their home on their behalf. We asked them to tell the AI what it should do, and began the writing prompt with, "OK, <AI name>..." to encourage respondents to communicate directly with the smart home AI. In the assistant-mediated scenario, we did not enforce a time minimum on the response page, and we did not provide the device or data list for reference, in order to discourage respondents from pasting parts of the list into the textbox. As in the unmediated scenario, we finished with demographic data collection and questions to assess technological familiarity.

3.2.2 Subjects

We submitted these two questionnaires to Amazon Mechanical Turk [131]. Since we planned to analyze the linguistic characteristics of the responses, we limited respondents to those located in five countries with large populations of native English speakers (United States, United Kingdom, Canada, Australia, and New Zealand) in order to increase the proportion of native English speakers in the eligible respondent pool.

Subjects were recruited to participate in an academic study about either Internet of Things applications or smart homes. The unmediated questionnaire was advertised with, "We are conducting an academic survey about Internet of Things applications. We want to understand how you think about and describe Internet of Things applications." The assistant-mediated questionnaire was advertised with, "We are conducting an academic survey about smart homes. We need to understand the way that you would talk to a smart home artificial intelligence." Workers who completed the unmediated questionnaire were compensated $0.80,

and those who completed the assistant-mediated questionnaire were compensated $0.40. The difference was due to the difference in the enforced time limit (the assistant-mediated questionnaire did not have a time minimum, so users spent less time on the task).

As shown in Table 3.1, we received 1,534 responses once we filtered out 53 bad actors who pasted parts of the prompt, URLs, or jibberish into the submission form. There could potentially be overlap between the participants in the unmediated questionnaire and the assistant-mediated questionnaire, but the two questionnaires were administered weeks apart and most workers had very few prior HITs, so we do not expect many duplicate participants.

3.2.3 Mental models indicators

Mental models are difficult to observe directly, so we approach the analysis of the questionnaire responses from multiple angles. We compare three different attributes of the prompts: qualitative differences, characteristic words, and the operation profiles.

Qualitative differences. We read all 1,534 of the responses, which was necessary to remove responses from obvious bad actors as described in Section 3.2.2. To guide our understanding of the structure of the responses, we also performed the exercise of sketching out a natural language grammar for a subset of the responses that included the top 100 most common nouns and verbs. We share some representative responses for each prompt in Table 3.2 to help the reader gain an intuition for the qualitative differences that underly the quantitative results.

Characteristic words and phrases. We used a χ^2 test to determine the keyness of each word used in each of the four response sets. Words with higher keyness are more suggestive of statistical differences between the response sets. This is a standard feature selection technique used in natural language processing to discover which words are particularly distinctive in a particular corpus compared to other corpora. For each word we calculated the rate of its occurrence per 1,000 words in the responses to each of the four prompts. Applying a χ^2 test to these rates produced the keyness values, which are simply the χ^2 statistic for each word. We display the rates and keyness values for the top 25 most key nouns, verbs, and other words in Figure 3.1.

Operation profiles. To paint a more rigorous picture of what users want to do in response to each prompt, we developed a set of labels to categorize sentences based on the operations defined in Section 3.1.2. These labels were "immediate action," "conditional action," "direct question," "indirect question," and "notification." After reading the responses to the unmediated prompts (see Table 3.2), we also added the labels "wants device," "wants remote control," "wants automation," and "wants to know." We also included a "none of the above" option for other kinds of sentences. Once we determined the set of labels that we were interested in, we took the first sentence of each response (for a total of 1,534 sentences) and asked three trained annotators to label the operations in each sentence, permuting the list of available labels that we displayed with each sentence to prevent biasing. The resulting Cohen's kappa statistics between the pairs of annotators were 0.76, 0.76, and 0.78, indicating a good level of inter-rater agreement. From the three sets of labels, we calculated the mean and standard deviation of

the percentage of the responses to each prompt that contain a particular operation. Since a sentence could have multiple independent phrases in it with different kinds of operations (e.g., "Turn on the light and tell me my weight"), the percentages of responses that contain particular operations are independent from each other and do not sum to one hundred. The resulting operation profiles are shown in Figure 3.2. These profiles give us a sense for what respondents to a particular prompt wanted to do in their hypothetical smart home.

3.3 Findings

Despite Mechanical Turk's reputation for skewed demographics, we found that our respondents were fairly representative of our ideal study population. Over 99% of respondents were from the United States. Respondents were slightly more male, with 58% male and 41% female. Most respondents were young but older respondents were still present, with 20% age 18-24, 43% age 25-34, 20% age 35-44, 10% age 45-54, and 6% age 55+. The distribution of educational attainment was skewed slightly higher than that of the US population [132]. Most respondents had earned at least a bachelors, with 34% having earned a high school degree, 50% having earned a bachelors (compared to 32% in the 2015 U.S. census), 12% having earned a masters, and 2% having earned a PhD (compared to 12% advanced degree holders nationally). However, despite higher levels of education, most respondents were not particularly knowledgeable about computer science. Only 9% of respondents categorized their occupation as computer worker, and 76% of respondents reported their exposure to CS concepts as either "low" or "none." 66% had never heard of the Internet of Things before.

3.3.1 Priming produces distinct mental models

As illustrated in Table 3.2, the answers we received to the four different prompts exhibited distinct qualitative differences, particularly between the unmediated and assistant-mediated prompts. Responses to the unmediated prompt are expressed as hypothetical situations about what the respondent *would like* in their home ("I would like" and "I would want" were the most common 3-grams), whereas the responses to the assistant-based prompt are expressed as executable directives ("turn on the" was the most common 3-gram for both the Device and Data prompts). The Unmediated Devices responses emphasize the devices that the respondent would like and why, whereas the Unmediated Data responses specify high-level "apps" that the user would like. While both assistant-mediated prompts include immediate actions like "turn on the lights" and conditional actions like "turn on the lights when I get home," the Assistant-mediated Data responses contain more questions and requests for monitoring and notification.

These observations are also reflected in the striking differences between operation profiles clearly visible in Figure 3.2. The Unmediated Devices responses are dominated by users expressing desires for devices, whereas the Unmediated Data responses show more requests for automation and information. The assistant-based prompts both have immediate actions and

Unmediated Devices	Unmediated Data	Assistant-mediated Devices	Assistant-mediated Data
I would definitely want the smart watch to control the majority of the devices and controls in the house. The controllable lights would be a nice feature to have. I would definitely look for the smart door lock and smart thermostat. Those are things that make life easier for the convenience factor. The smart car would also be a good investment. Not only is it easier to use, it is energy efficient. Motion sensors in each room would save on electricity. [...]	I would want interface between my security system, smoke alarms, CO alarms, and cell phone. I would also want to be able to control the climate control systems (A/C and heat) from my cell phone, and monitor the temperature. It would also be nice if I could see my electricity usage in realtime, and customizable alerts sent to my phone would be quite helpful.	Set the temperature to 70 degrees. Lock the door. Close the blinds. Fetch and read my email.	Please turn the red lights on dimly in my bedroom and start playing Marvin Gaye music. Dim all the other lights in the house.
First and foremost, I would love to have a controllable TV that is hooked up to my video library. This would be not only an incredible time saver, but also a space saver as well. A temperature sensor on my smart phone that is hooked up to my thermostat would also be beneficial, as a traditional thermostat only takes into account the temperature near that thermostat. Along with the controllable TV, "smart" speakers that are hooked up to my music library would be nice. In fact, an app that turns my smart phone into a universal remote control might be my favorite "Internet of Things" application. It would offer tremendous convenience, as my smart phone is always within arms-reach. [...]	I would like an app that tracks my heart rate along with the calories burned, sleep cycles, activity, whether I was calm or not. [...] I would like to have one where I can see my child and watch the rooms where my caregiver is. Knowing where they are, or what they did. I wouldn't be using this all the time as I trust my caregiver, but at times I would like to see how much time is spent in specific room, such as the living room. Or to track what TV apps (like Hulu or Amazon prime) was run on my TV and for how long. Maybe what was watched and the length of time. [...]	Please wake me up at 9:00 am with some pleasant music. Please make coffee at exactly 10:00am. Please set the alarm before I leave the house. Please water the yard at noon for 20 minutes. Please turn on the porch light before 7pm.	I would like you to make sure that when I leave the house, all lights, AC, and electronics are turned off and the door is locked. While I am gone, I would like you to monitor the house, and call my phone if anything strange happens (anyone enters the house, any objects are moved, etc.). In addition, I would like to use your knowledge of transportation to plan when to leave the house to catch the bus to work. You can also alert me to any poor weather conditions before they arrive. Thanks!
I think it'd be cool to be able to put a sensor at the end of the driveway so that when certain cars drive up it will open up the garage doors [...]	I would like the ability to know how much water, electricity, and gas I use, with a running ticker of how much it is costing me. I would also like a breakdown of which rooms/objects are using the most. I would also like to know what lights are on, and if there is a window open in a room that is running heating or air conditioning. [...]	Lower the lights, lock the doors, begin to play Madonna's last two albums on shuffle from every speaker. Also synchronize the house RGB lights, my clothing lights and vibration patterns to the rhythm and tone of the music.	Tell me my electricity consumption and gas consumption. How has my sleep been lately? When do I wake up? Am I exercising enough?
		Start my car and turn on the heat and radio. Open the garage. Lock the house doors. Adjust the temperature on the thermostat. Check all appliances and make sure they are off. Close garage after I drive away.	Who is in the home with me? Where is my car and how fast is it going? How is my heart rate?
		Please be sure the door is locked and the temperature is set at 75 degrees. Next, be sure the volume is set on #15 as I want to listen to some music.	I want you to turn on the lights when I walk into every room and play music whenever I am in the mood for it.
		Turn on my air conditioner 30 minutes before I get home at 4:00 pm.	Turn on the dishwasher when I get home from work so I can serve dinner, and make sure to alert me via a text message a half hour prior to my wife getting home so I have time to put the finishing touches on the meal.
		Turn TV on. Lock all doors. Adjust temperature to 70 degrees Celsius.	

Table 3.2: Example responses for each interaction framing. This table shows several responses for each framing, with individual responses separated by blank lines. These examples provide an intuitive feel for how the responses differ between prompts.

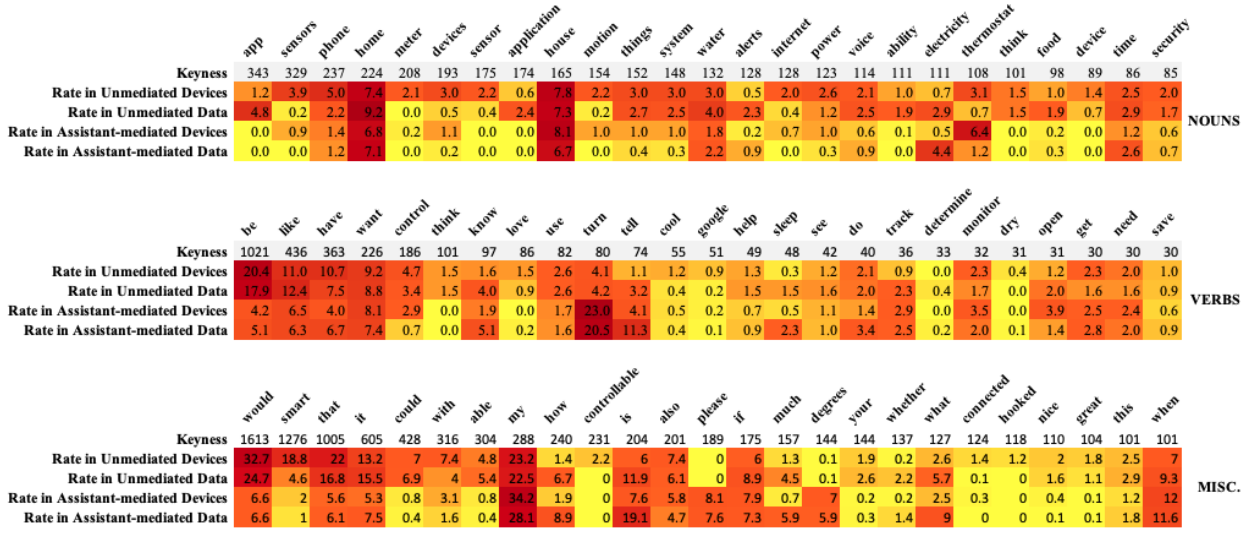


Figure 3.1: Characteristic nouns, verbs, and miscellaneous words for the different interaction framings. This figure shows the top 25 nouns, verbs, and remaining words sorted by keyness as determined by their χ^2 statistic. Words with higher keyness are more suggestive of statistical differences between the response sets. The rates show how often each word occurs per 1,000 words within each set of responses. Differences between prompts can be identified when the rates vary within a column. Notable outcomes here are that the Unmediated Devices responses are characterized by the words like "sensor[s]", "device[s]", "meter", and "things," with an emphasis on "control" and "controllable," through a "phone." This suggests users were focused mainly on remote control of devices through a phone. The Unmediated Data responses, on the other hand, emphasize "apps" and "applications," as well as "alerts" and "know[ing]," though they also still score highly on "control" and words associated with conditional actions and notifications like "if" and "when." This supports the notion that this conceptual model will encourage users to express various kinds of requests for information and automation in addition to remote control commands. The two assistant-based conceptual models showed mostly similar word rates, suggesting that placing an intermediary between the system's devices or data capabilities smoothes out some of the differences seen in the two unmediated prompts. Both showed high rates for "please" and "turn." However, for the assistant-mediated data prompt, there was more of an emphasis on "tell" and less on "control", a mental shift reflected in other query-related words like "know," "how," "much" and "what."

conditional actions, but the Assistant-mediated Data prompt also shows a large proportion of questions and notifications. Presenting the same smart home system in four different ways resulted in four distinct operation profiles.

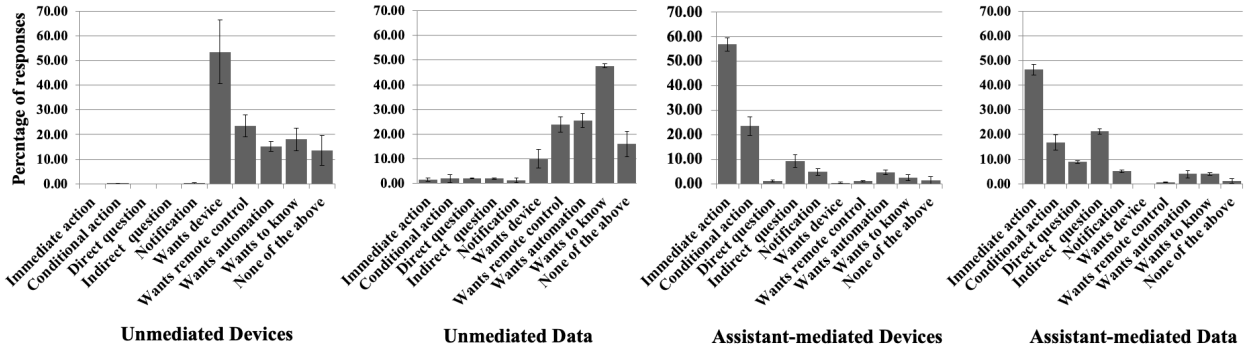


Figure 3.2: Operation profiles found in the four sets of responses. These charts show the proportion of different kinds of operations in the user responses to the questionnaires. We took the first sentence of each response (for a total of 1,534 sentences) and asked three trained annotators to label each sentence from a set of labels that we provided based on a qualitative exploration of the dataset. They could provide multiple labels if there were multiple phrases, so the percentages are independent and do not add up to 100. One notable observation is that the operation profiles are all different, demonstrating that the interaction framings had a priming effect on our users. When combined with qualitative analysis and the words and phrases associated with each prompt, we can get a picture of the mental models behind each of these distributions.

3.3.2 Unmediated Devices encourages the least automation

The Unmediated Devices framing is a popular one, used by the device-centric smartphone apps that come with every smart device. However, this framing limits user expectations of interoperability and automation. When presented with the Unmediated Devices framing, users seem not to see "smart devices" as connected devices, instead tending to treat them as isolated islands of functionality.

In response to a prompt that said, "[describe] different *applications* that you would want in your smart home," the majority of users expressed a desire to have a *device*. The operation profile for responses to the Unmediated Devices prompt, as shown in Figure 3.2, is dominated by "wants device," with 53% of sentences expressing a desire for a particular device or devices. The example responses provided in Table 3.2 give some intuition for how users conveyed this, with sentences like, "I would definitely want the smart watch[...]" and "I would love to have a controllable TV." In Figure 3.1 you can see that particularly characteristic of responses to this prompt are words like "sensor[s]", "device[s]", "meter", and "things," with an emphasis on "control" and "controllable."

The word "phone" is particularly associated with this conceptual model as well. The overall picture that emerges is that users responded to the Unmediated Devices prompt by expressing a desire to have controllable smart devices that the user can control manually, primarily with their phone, with surprisingly few automatic applications running on the smart home system.

These results suggests that when presented with the Unmediated Devices framing, users tend to think of the device itself and the application behavior as one and the same. The relative lack of higher-level applications that run across multiple devices suggests that respondents to this prompt did not tend to perceive a global computational scope or a sense of general programmability in their smart home.

3.3.3 Unmediated Data encourages the most automation

Unmediated Data is not a very common interaction framing in smart spaces, best corresponding to dataflow wiring interfaces for automation, such as Node-RED [133]. However, this interface shows a lot of potential for encouraging automation and information acquisition, while still supporting remote control interactions.

Users presented with the Unmediated Data framing showed a propensity for higher-level automation applications compared to the Unmediated Devices framing. Figure 3.1 shows that the words "app" and "application" were particularly associated with responses to this prompt. The operations profile in Figure 3.2 shows that Unmediated Data respondents expressed more desire for automation than the respondents of any other conceptual model (corresponding to the "conditional action" label in the assistant-mediated prompts).

However, the majority of sentences were labeled as "wants to know." Many of the applications described in the Unmediated Data responses focus on monitoring and providing alerts to the users. You can see this in Figure 3.1 where the words "alerts," "know," "see," and "track" were found to be strongly associated with this prompt. Also ranked highly were measurable phenomena, like "water," "electricity," and "sleep". Users also wanted to be able to solicit information from the system on demand, as evidenced by the relative keyness of "what," "is," "how," and "much." Another significant word was "if," which is associated both with trigger-action automation as well as with notifications and alerts.

We call this mental model *watchdog* to convey that users tended to think of the system as a global observer whose primary job is to monitor and inform users, and whose global scope gives it the ability to run integrated automation applications across the system.

3.3.4 Adding an assistant to devices increased automation

The operation profile of the responses to the Assistant-mediated Devices prompt shown in Figure 3.2 reveals that most responses (57%) were "immediate actions" (e.g."turn on the lights"), corresponding to 23% of "wants remote control" in the Unmediated Devices framing. However, more importantly, 24% responses contained conditional actions describing automation, compared to just 15% of "wants automation" for Unmediated Devices. Adding the assistant personification on top of the collection of smart devices therefore helped users better imagine automation scenarios.

Users were polite, as "please" was the 20th most common word, and "I would like you" and "would like you to" were the second and third most common 4-grams respectively. This politeness suggests that the users perceived the system to have social agency.

An interesting pattern that occurs not just here, but in both assistant-based prompts, is providing a reason or goal after specifying actions to take. One example from the Assistant-mediated Devices prompt is, "adjust the blinds every morning around 5, so that I wake up easier when my alarm goes off." Other reasons that users expressed to explain actions were things like, "so I can save energy," "so I can watch Netflix," "so I can go to sleep," and "so I can let [someone] know [something]." Providing goals as a part of instructions suggests that users potentially expect assistants to understand basic goals.

3.3.5 Assistant-mediated Data elicits control and questions

Like the Assistant-mediated Devices prompt, the Assistant-mediated Data prompt produced mental models that supported primarily immediate actions (46%) with some conditional actions (17%), but the operation profile shows that unlike in the responses to any other prompt, 39% of the operations were labeled as questions (both direct and indirect) or other requests for information. While requests for information have seldom appeared in previous research on smart home mental models, it is clear that this underrepresentation in the literature is not out of a lack of user interest in queries as a form of interaction.

One interesting observation is that despite the strong presence of questions in the Assistant-mediated Data response set, the word "why" does not appear a single time in any of the Assistant-based responses (but does appear in responses to the Unmediated prompts). This suggests that users do not think that assistants are capable of either introspection ("why did you turn on that light?") or explanation ("why is my energy bill so high?"). It is possible that this is due to the limitations of our method of data collection, which did not involve repeated interactions with the assistant. These questions might arise if users lived in a smart home and interacted with an assistant over a long period of time. Nevertheless, it is interesting that not a single respondent thought to ask an assistant "why."

3.3.6 Priming strength differs by age and technical background

Prior work has shown that occupants interact differently with smart homes depending on their technical expertise and age. Technical occupants often assume the role of programming and maintaining the system, whereas non-technical occupants communicate their wants and issues to the technical member for translation into programmatic instructions [24]. Ethnographies have also shown that older occupants and younger occupants may have different needs and expectations from smart home interfaces [134].

We analyzed demographic subpopulations to better understand differences in the way these populations respond to conceptual models. We determined technical expertise by whether the respondent self-reported their exposure to computer science concepts as None ("No exposure to ideas of computer science"), Low ("Some exposure to computer science concepts"), Medium ("Undergraduate computer science student"), or High ("Computer science graduate student or professional"). When determining age, we define older occupants as those whose age is 55 or older, and younger occupants as those who are 34 or younger. We used a χ^2 test on the

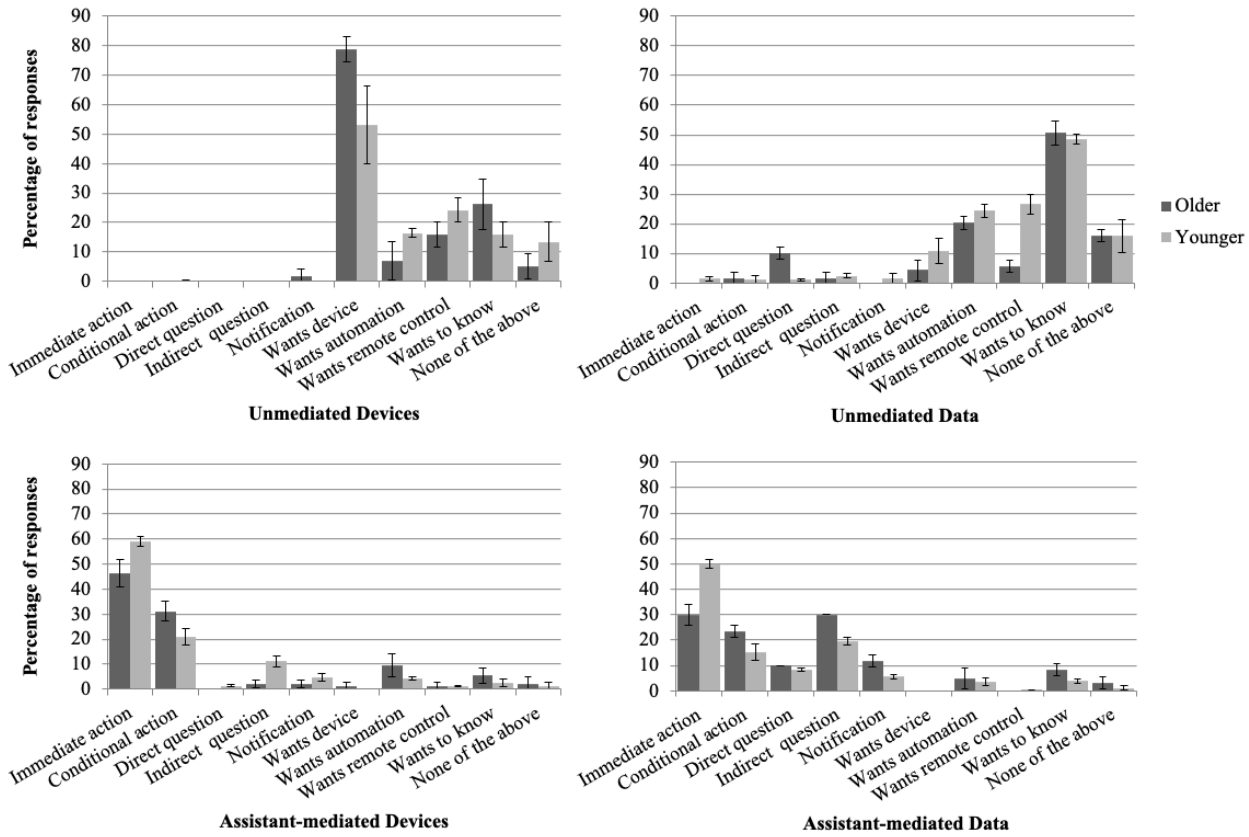


Figure 3.3: Comparison of operation profiles for older users vs. younger users.

observed frequency of operations in the populations' responses to each prompt to determine whether any of the operations significantly differed between the two populations we compared.

Figure 3.3 shows that older respondents respond more strongly to priming than younger respondents, whereas younger respondents tend to adhere less to the primed model. Given the Unmediated Devices conceptual model, older respondents were significantly more likely to want a device ($p = 0.025$), and in the Unmediated Data responses, older respondents were more likely to ask direct questions ($p = 0.007$). Conversely, when presented with an Unmediated Data conceptual model younger respondents were more likely than older respondents to want remote control capabilities (0.0003). Younger respondents were also more likely to ask indirect questions in the Assistant-mediated Devices responses despite the device framing ($p = 0.014$), and express immediate actions in the Assistant-mediated Data responses despite the data framing ($p = 0.026$).

Figure 3.4 tells a somewhat similar story. While there are no significant differences between respondents with high CS exposure and no CS exposure when presented with the familiar Unmediated Devices conceptual model, the populations diverge when presented with the remaining three less-familiar models. Given the Unmediated Data framing, those with high CS exposure are more likely to recognize the ability to perform automation ($p = 0.037$),

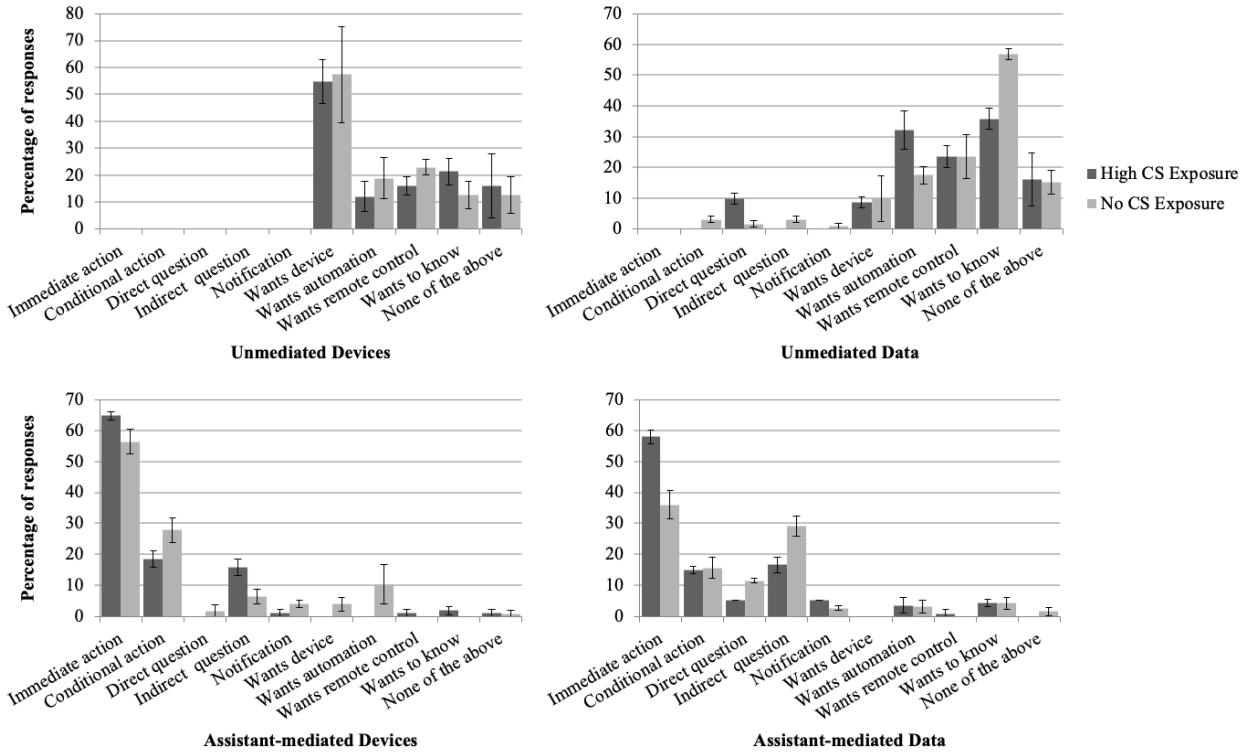


Figure 3.4: Comparison of operation profiles for users with high computer science exposure vs. no computer science exposure.

whereas those with no CS exposure are much more likely to stick with expressing a general desire to know things ($p = 0.029$). Presented with the Assistant-mediated Devices prompt, respondents with high CS exposure were more likely to say indirect questions, whereas those without the training would still say they wanted a device ($p = 0.0463$) or automation ($p = 0.001$). Finally, when presented with the Assistant-mediated Data framing, those with high CS exposure were significantly more likely to request that the system perform immediate actions ($p = 0.0234$). As with older vs. younger respondents, those with no CS exposure tended to conform closely to the primed conceptual model, whereas those with high CS exposure were able to invoke functionality that did not conform as closely to the model. The similarity between these results is not due to correlation between younger age and more computing expertise, as those who self-reported higher CS exposure tended to be older.

3.4 Implications for smart space system design

Our findings about the effect of priming on users' mental models have significant implications for mental model research and system design.

3.4.1 Comparing interaction framings is critical

Researchers studying mental models for the purpose of building intuitive end-user interfaces need to be aware of the priming effects of system interaction framings. Researchers must assume that the interaction framings used in the study scenarios will prime users, and that is a good thing because so will the interaction framings used to present the actual system. Researcher will need to choose interaction framings consciously in a principled way, so that the mental models and operation profiles resulting from different interaction framings can be compared. Building up a corpus of conceptual models and their associated mental models and operation profiles will be beneficial for system designers who want to know what interaction framings they should select without having to spend too much money or time running their own user studies.

System architects of smart spaces should also embrace priming as a part of their system design space. The interaction framings used to represent the system will strongly influence the kinds of interactions that users will task the system with. This means that higher-level choices about the way the system is presented to users will have lower-level consequences for the system requirements. System engineers will need to become aware of the coupling between the broad space of interaction framings available to represent the system and the operational implications of each option.

3.4.2 Consider how subpopulations may be primed differently

Designers will also need to keep their target audience in mind. The operation profile is not just a function of the framing, but also the subpopulation interacting with the system. Technically-inclined users may have a deeper understanding that allows them to discover more functionality than the conceptual model conveys, whereas non-technical people will be much more likely to only express those desires that can be clearly satisfied within the confines of the presented model. Similarly, older occupants will also be greatly influenced (and constrained) by the conceptual model.

3.4.3 Consider automation when choosing a framing

Currently, systems exist that do not support interoperability between devices very well, or which feature single-purpose devices. For those systems, the Unmediated Devices framing may be a sensible choice of framing, as manual one-on-one interactions with individual devices are what the system can most easily support. However, the value of ubiquitous computing is projected to come from the ability to run integrated applications on general-purpose physical computing systems. In transitioning from a special-purpose to a general-purpose future, and as the number of devices scales up, it will be important to keep in mind other interaction framings that we could transition to away from Unmediated Devices that would encourage users to think more broadly.

Intelligent assistant interfaces are another popular smart space interface that we show have advantages over Unmediated Devices in the long term, in that they help users imagine more potential for automation. However, intelligent assistants may carry additional user expectations that may be hard to satisfy in practice. Prior work has shown that to manage user expectations of intelligent assistants, we need a way to convey to the user what the assistant understands [135].

Since system designers may reasonably wish to avoid the implementation complexity that comes with users' expectations of assistants, it is significant that our findings showed that the Unmediated Data framing also encouraged users to think of high-level automation applications, presumably due to the global scope that "monitoring" data implies. Data-centric interaction framings have been relatively unexplored in smart spaces, and represent a potentially fruitful area of future work.

3.4.4 Incorporate artificial intelligence primitives

Finally, we found that end users will comfortably use primitives that have traditionally fallen under the umbrella of AI rather than systems. Previous work has identified that in the smart home domain users will employ "fuzzy triggers" which specify qualitative preferences like "comfortable," "normal," and "sufficient" that must be learned by the system [75], but we identify two new concepts in user applications that would require artificial intelligence techniques to implement:

Prediction. Some commonly-occurring commands and questions in our dataset require prediction to execute. Examples include actions relying on the condition "before," such as "have coffee ready before I wake up," and questions starting with "when will," such as "when will my husband be home." The basic nature of these primitives suggests that machine learning for prediction tasks should be included in smart space systems at a fundamental level.

Goals. Many responses to the assistant-based prompts describe a goal and provide example actions in order to demonstrate to the assistant how to achieve the goal. If the system were able to understand goals and planning, it would provide the assistant with a great deal of flexibility in achieving the specified goals. In the example "adjust the blinds every morning around 5, so that I wake up easier when my alarm goes off," the system could detect that the user would like help in waking up and potentially suggest (or attempt) supplemental strategies, like turning on the lights to full brightness, or playing additional sounds. The ability to comprehend even basic goals would give a smart space power to improvise and make suggestions to aid its occupants. Systems that wish to support intuitive end-user programming may therefore benefit from drawing on the extensive work done in artificial intelligence on goal-based assistants, planning, and learning from demonstration.

3.5 Summary

In this work we show that the interaction framings used to present a smart home system to end users have a significant priming effect on users' mental models and the kinds of interactions that users expect to have with the home. We introduce a preliminary framework that identifies several dimensions along which conceptual models can be described, and then focus particularly on interaction framings situated along two of those dimensions: the system capabilities (devices vs. data), and the system personification (unmediated vs. assistant-mediated), the combination of which results in four archetypal interaction framings. We then characterize the mental models that users form when presented with those interaction framings. We use Amazon Mechanical Turk to collect over 1,500 responses to questionnaires where we describe a single hypothetical smart home's capabilities using one of the four interaction framings. To gain insight into users' mental models, we analyzed the qualitative differences between the sets of responses, as well as differences in the types of tasks the users ask the system to perform, and the words characteristic of each prompt according to statistical tests. Based on our analysis of these three signals, we find that different interaction framings strongly affect the mental models the users have of the system and the ways that they want to interact. We also found that older users and users without computer science expertise tend to respond more strongly to priming than younger users and users with computer science expertise.

Our findings have major implications for both HCI researchers and system designers. HCI researchers studying mental models will need to be aware that there is no unbiased way that end users think about these systems and that priming should be embraced as an omnipresent influence on mental models, requiring researchers who are trying to understand mental models to do comparative evaluations of different system interaction framings. System designers should consider the choice of framing to be part of their design space, since it will heavily influence the kinds of operations that the system will need to support.

We also found that end users employ computational primitives that have traditionally fallen under the umbrella of artificial intelligence, such as prediction and goal-oriented planning. Consequently, HCI researchers, system designers, and AI researchers should take active measures to bridge the institutional gaps between them and collaborate closely on future platforms.

Finally, we found that the most popular framing used in smart space interfaces today, the Unmediated Devices framing that corresponds to device-centric smart phone apps, produced the most limited kinds of interactions, characterized by manual one-on-one interactions with individual devices. High-level applications like automation and queries often do not even occur to users presented with this framing, though they do occur when users are given different interaction framings of the same system. This suggests that as smart space architecture continues to evolve towards interoperable general-purpose physical computing systems, we should move away from the Unmediated Devices framing if we want end users to think of valuable, integrated applications that operate across a network of devices.

The limitations of the Unmediated Devices framing could be mitigated merely by adding

an intelligent assistant mediation layer on top of the devices, which increased the amount of automation possibilities that occurred to users. This corresponds to another popular interface in modern smart homes, intelligent assistants, such as those operated through smart speakers. Due to its advantages for mental models and current popularity in modern smart spaces, the interface designs for discovery that we introduce in the latter half of this dissertation will focus on an intelligent assistant framing.

Chapter 4

How Reliance on Proper Names Impairs Smart Space Discoverability

In the last chapter, we examined the *device-centric* aspect of the proper device name paradigm and saw how it limits user expectations of system capabilities when not mediated by a intelligent assistant. In this chapter, we are going to look at the *proper name* aspect of the proper device name paradigm and characterize the degree to which proper device names are not guessable, making it difficult to operate unfamiliar smart spaces without discovery aids. As early as 1987, Furnas identified the "vocabulary problem" for language-based interfaces [76], stating that even the best-possible canonical name for an information object will still be hard to guess. Here, we examine the degree of that problem in the smart space domain.

4.1 Guessability Study

To measure how difficult it is to guess proper names for smart space devices, we want to determine what people think the most guessable names are for devices in a smart conference room and see whether there is consensus or disagreement on the names. Prior work has devised methods for maximizing the guessability of names, calculating formal measures of the guessability of a particular name, and calculating the agreement of a set of proposed names [136]. We use the latter metric in our analysis of the collected set of proposed names.

4.2 Methodology

We ran a Mechanical Turk study to collect guessable names for devices in a hypothetical smart conference room. We released two different tasks that differed in their visual prompt, shown in Figure 4.1. The first prompt showed each worker a picture of the conference room with all of the lights highlighted, and asked the worker to give a unique name to each light (Figure 4.2). The second prompt showed the same image, but highlighted all of the non-light devices. We split the devices across two prompts to reduce visual clutter.

(a) Lights (b) Non-lights

Figure 4.1: Prompts indicating smart devices and device groups. The study was split into two prompts, one for lights and one for non-light devices, to reduce visual clutter.

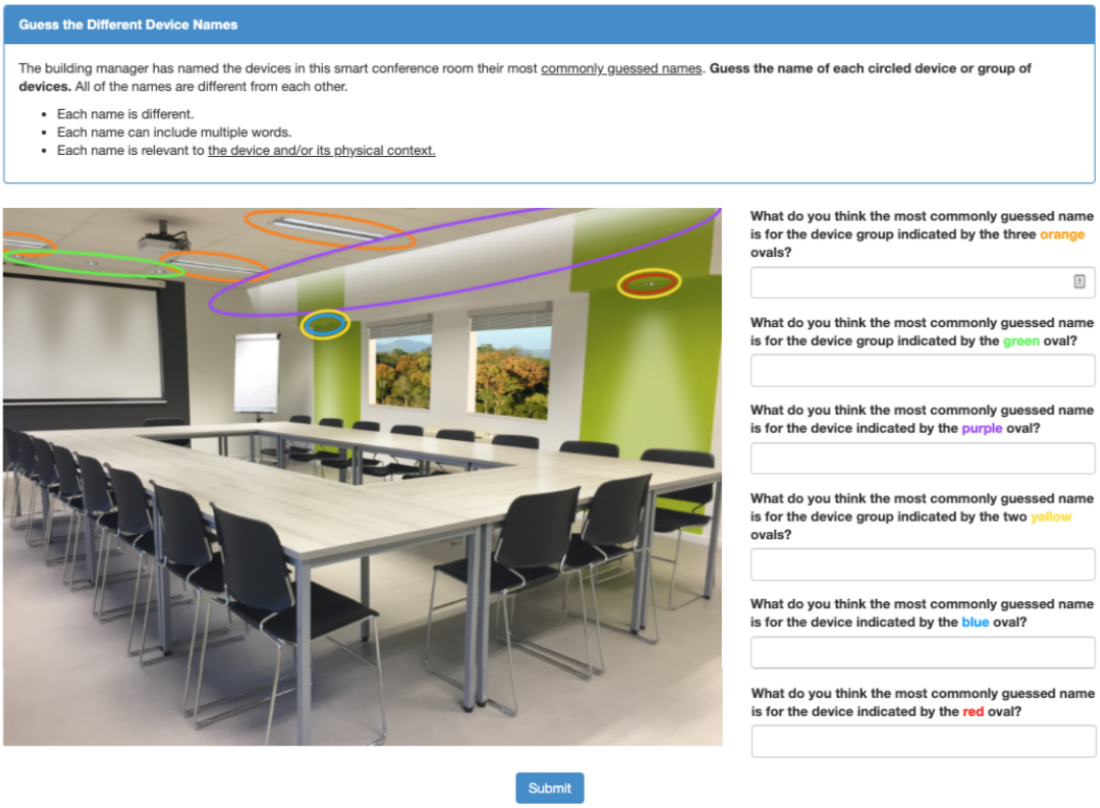


Figure 4.2: Mechanical Turk prompt for the guessability study. Workers were shown one of the two pictures of the simulated conference room, with devices and device groups circled in colored ovals, and asked to provide what they thought the most commonly guessed names were. Each name had to be different than the other names.

The smart conference room had six lights (or groups of lights): overhead fluorescent lights, a set of front lights over the projector, a pair of accent spotlights (which included an independently operable right light and left light), and a recessed ambient wall light. There were four non-light devices or device groups: the blinds, which included an independently operable left blind and right blind, and a projector.

In early pilots of this study where the instructions simply solicited candidate names, respondents often provided highly unique and creative responses, so our final study prompt emphasized that the devices had been assigned commonly guessed names and we asked participants what they thought that commonly guessed name was. The results of early pilots also led us to clarify that each name was different.

4.2.1 Data Analysis

In total, we collected 100 guessed names per device (or device group). We ran the names through a spellchecker to avoid purely orthographic differences that are unlikely to be an issue during smart space operation. To analyze the data, we generated histograms of the 100 names provided for each device name, as well as individual words in the names.

For each device we also calculated the agreement metric for the set of provided names, as described in the paper "Maximizing the Guessability of Symbolic Input" [136]. Since we calculated agreement for one device (what they call a "referent") at a time, the formula we used for each device is:

$$A = \sum_{P_i \subseteq P_r} \left(\frac{|P_i|}{|P_r|} \right)^2 \cdot 100\%$$

P_r can be considered the set of all respondents, and P_i the subset of all respondents who gave the same name i. In our problem formulation, $|P_r|$ is always 100, and $|P_i|$ is the number of times that the name i occurs in the set of 100 guesses. The possible values of agreement range from 1% to 100%. This metric is meant to be higher when more people provide the same name, reflecting the peaks of popularity, and meant to be lower if there are more unique names, reflecting the length of the histogram tail.

4.3 Findings

Consistent with other domains studied in prior work, we found that proper names for the smart conference room devices were largely not guessable. However, we also found that the precision problem for the smart space domain is acute, meaning that the same popular name could potentially apply to multiple devices. We also made a number of additional observations about how respondents named devices that can provide insight into how people conceptualize smart devices' relationships to each other and to the environment.

Spotlights	Wall light	Front lights	Left light	Fluorescent lights
75 unique names	**65 unique names**	**64 unique names**	**72 unique names**	**33 unique names**
2.54% agreement	**4.04% agreement**	**2.88% agreement**	**2.02% agreement**	**10.26% agreement**
11 spotlights	16 wall	8 spotlights	5 spotlight	21 lights
4 wall lights	5 recessed lighting	7 recessed lights	5 speaker	18 fluorescent lights
4 recessed lights	3 sky light	6 front lights	5 left light	9 overhead lights
3 overhead lights	3 recessed lights	5 recessed lighting	4 can light	7 fluorescent light
2 spotlight	3 ceiling	4 spotlight	3 light	7 ceiling lights
2 speakers	3 back light	3 presentation lights	3 left spotlight	6 fluorescent lighting
2 side lights	3 ambient lights	3 overhead lights	2 sprinkler	3 light
2 recessed lighting	3 ambient lighting	3 can lights	2 overhead light	3 ceiling light
2 lights	2 soffit	2 projector lights	2 lights	2 long lights
2 led light	2 lighting	2 lights	2 left recessed light	...24 more names
2 can lights	2 light	2 light	2 left accent light	
...64 more names	2 accent lights	2 circle lights	2 front wall light	
	...53 more names	2 can light	2 front spotlight	
		...51 more names	2 front light	
			2 camera	
			...57 more names	

Right light	Blinds	Left blind	Projector	Right blind
73 unique names	**36 unique names**	**31 unique names**	**14 unique names**	**39 unique names**
1.96% agreement	**8.54% agreement**	**11.06% agreement**	**58.06% agreement**	**6.20% agreement**
5 right light	20 blinds	27 blinds	75 projector	13 blinds
5 recessed light	12 windows	11 left blind	13 overhead projector	12 right blind
4 light	10 window blinds	8 shade	...12 more names	10 shades
3 spotlights	7 window shades	8 blind		7 blind
3 spotlight	7 window	6 left blinds		6 window blind
3 speaker	7 shades	5 left window shade		6 right blinds
3 right spotlight	5 window coverings	4 window blinds		4 shade
3 can light	2 window treatments	4 shades		4 right window shade
3 back light	2 window covers	3 window shade		3 curtain
2 right recessed light	2 wall	2 window blind		2 window shade
2 right accent light	...26 more names	2 left shade		2 window blinds
2 lights		...20 more names		2 right window blind
2 back wall light				2 right shade
...60 more names				2 mini blind
				...25 more names

Figure 4.3: Metrics and counts of names. For each device, we list the number of unique names in the set of 100 names participants guessed, and the agreement metric for the set of names. The agreement metric reflects both the popularity of the most popular names, as well as the number of unique names (more unique names suggests less agreement). Underneath the summary statistics, we list the names ordered by number of guesses, omitting any names that were only guessed once—the vast majority of submissions. Though guesses were reasonable, there was low agreement and high uniqueness, illustrating a fundamental guessability problem.

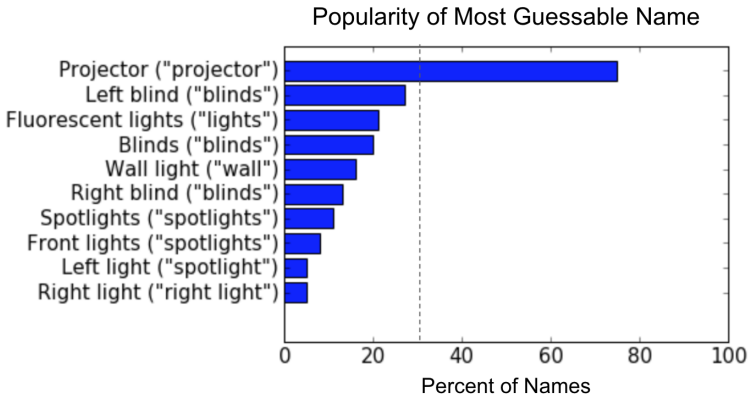


Figure 4.4: This chart shows the popularity of the most guessable name for each device. On the left, you can see the name we use to refer to the device, followed by the most popular name for the device in quotes. The bar shows what percent of the 100 names were that name. Except for the projector, all of the names are less than 30% of the submitted names (shown by the dotted line). This means if these devices were named these best possible names, a new user would still have between a 70-95% chance of being wrong on their first guess.

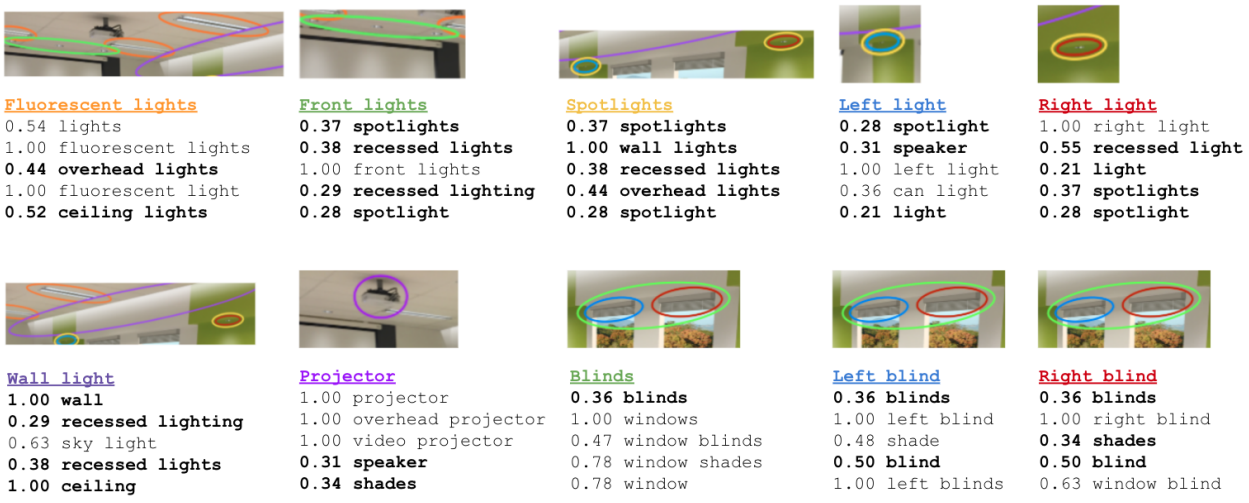


Figure 4.5: Top names and their precision. The underlined names in color are the names we use to refer to the devices in this chapter. Below, you can see the top five names submitted for each device and how precise they are, given by the odds that any two occurrences of the name in the dataset refer to the same device. The bolded names appear in the top five names of at least one other device. (We considered "wall" and "wall light," and "ceiling" and "ceiling light" to be the same, though when calculating precision they were treated as two unique names.) Note that for "Spotlights," any of its top five names could also refer to other devices in the room. The average precision of these top five names is 0.58, suggesting that if presented with one of these names, on average a user would have a 58% chance of matching it to the right device.

4.3.1 Device names were mostly not guessable

Only one device could be considered guessable—the projector. 75% of respondents guessed the exact name "projector," and a further 13% guessed "overhead projector." A full 94% of guesses contained the word "projector" somewhere in the name.

However, the most popular names for the remaining devices were only provided by 5-27% of the participants. This means that even if the smart devices were named the best possible names, there is a 73-95% chance of the first guess failing. This lack of guessability means that we cannot just name devices better—we need a means of discovering proper device names.

4.3.2 Popular names could apply to multiple devices

Another potential discovery issue, highlighted in Figure 4.5, is that the most guessed names for one device were often among the most guessed names for other devices. For example, "recessed lighting" appeared as one the top names for four of the lights, making it a popular but highly ambiguous name. This overlap between top names makes it difficult to support the infinite aliasing or synonym-based solution to naming recommended by Furnas.

Furnas referred to this the "the precision problem," but found that the degree of the problem varied greatly between domains. We find it is a major concern in the smart space domain, where there are similar types of entities with relatively few distinguishing features. Merely providing users with a list of smart device names will not suffice to support one-shot interactions, as users may not be sure which name refers to which device. Interfaces must help users ground each name to its real-world device.

4.3.3 The order of naming in hierarchical device groups matters

The order in which users name entities affects what names they choose. We found that participants tended to start with most general name first (e.g., "light" or "blinds"), then became more specific if necessary for subsequent guesses. For example, if the left blind was named before the group of all blinds, the left blind would likely be named "blinds" and the group of all blinds would be named a synonym, such as "shades," or with an additional adjective, such as "window blinds." In the final iteration of the guessability study we asked people to name the most generic groups first.

4.3.4 Names do not always capture device relationships

Relatedly, we also found that people did not preserve hierarchical relationships or type information in their naming schemes. If the group of two spot lights together was called "spot lights," the right spot light might be called "can light" and the left spot light might be called "recessed light," losing the type information, the group hierarchy information, and the spatial relationship information. This may be because such relationships were not noticed. The prevalence of this kind of naming scheme (or lack thereof) suggests that there is

a potential mismatch between the flat device hierarchies that typical users perceive on their first impression of a space and the tree-like type or group structures of the actual system. An interesting open question is whether users would more quickly perceive type, group, or spatial relationships if presented with a naming scheme that reflected them.

4.3.5 Frames of reference can differ

For device names that included location information, users used different physical frames of reference or coordinate systems. While participants appeared to agree that the lights over the projector were in "front," and that the blinds were "left" and "right," there was disagreement over whether the spot lights in front of the green stripes were "left" and "right" (19% of names) or "front" and "back" (12% of names). These differences depend on whether the participant considered the origin to be the windows or the projector screen. This complements the results of Brumitt and Cadiz [87], which found that the reference point for relative locations in a smart home living room was always with respect to facing the television screen.

4.4 Summary

In this chapter we ran a guessability study to determine how fundamentally difficult it is to guess the proper names of smart space devices. We found that for most devices, even the most popular, best-possible names are not particularly guessable, resulting in a 70-95% chance of the user guessing the name wrong in the first interaction. This highlights the need for a lexical discovery method. We also found that even given a list of device names, it may still be difficult for the user to guess the correct name in the first interaction, since the top names could refer to multiple devices. This means we additionally need a grounding discovery method that helps map names to devices.

These results highlight the unique discovery challenges facing smart spaces that are designed around proper device names. In the next two chapters, we will explore interface designs that can help users overcome these challenges to achieve one-shot interactions in unfamiliar smart spaces.

Chapter 5

Improving Smart Space Discoverability with Autocomplete-Enabled Assistants

In the last two chapters, we characterized several smart space discoverability issues caused by the "proper device name" paradigm. We found that an intelligent assistant interaction framing can help mitigate some of the limiting effects of device-centrism, and we also found that proper device names are fundamentally not guessable. In other words, we cannot just name devices with better names, we need a discovery aid, ideally framed as part of an interaction with an intelligent assistant. These findings will help guide our solutions to help users perform one-shot interactions in unfamiliar spaces.

Over the next two chapters, we center our discovery-oriented interface designs around a messaging approach for smart space intelligent assistants. By messaging assistants, we are able to use visual discovery mechanisms such as autocomplete to help users understand what can be done in the room and how to invoke those actions using proper names.

An autocomplete mechanism for smart spaces must satisfy different objectives than it does in more traditional autocomplete applications like search. Since smart spaces are multi-modal with many different potential interfaces to the same system, the most effective discovery mechanism would not only help the user perform tasks, but also help them gain knowledge that can serve them later when performing tasks using other interfaces, such as graphical remote control apps, voice interfaces, text interfaces with no autocomplete, and even physical switches. In essence, the goal of autocomplete in an *unfamiliar* smart space is to make it a *familiar* smart space, and enable the user to be able to stop using autocomplete as soon as possible. The autocomplete interface should therefore aid the user in quickly and accurately accomplishing their task, while also building up subconscious semantic and lexical knowledge that can transfer to other modalities, where autocomplete or other discovery mechanisms may be unavailable.

With these goals in mind, we propose three different design dimensions for autocomplete where the best choice for the smart space domain may not be the same as the best choice

for more common autocomplete domains like search. Within these three dimensions, we evaluated eight different autocomplete designs, taking a large-scale methodological approach. We developed an online, unsupervised evaluation platform and deployed it on Mechanical Turk in order to collect performance metrics and gather a diverse array of feedback for our nine interfaces. Mechanical Turk allowed us to quickly gather 246 responses. While simulation-based studies may fail to capture important behaviors that only manifest in an embodied context, we were able to collect performance, learning, effort, and satisfaction metrics, as well as qualitative feedback, across a broad range of participants.

We find that all autocomplete users were better able to accomplish tasks and learn more about the semantic and lexical capabilities of the system than users of the baseline interface with no autocomplete. However, we found few differences between autocomplete designs. This is likely due to the fact that all autocomplete users still had to make many trial-and-error attempts to ground each proper name to its real-world device, thus learning comprehensively about the overall room during their attempts. We conclude that while autocomplete takes us a step closer to our goal, the grounding problem must still be solved to achieve one-shot interactions in unfamiliar smart spaces.

5.1 Messaging Smart Space Assistants

While most users currently interact with smart space intelligent assistants using voice, Google Assistant, Siri, and Alexa all support some degree of text-based interactions as well. By messaging smart space intelligent assistants, we can take advantage of the visual medium to use autocomplete suggestions as a mechanism to address lexical and semantic discovery, while maintaining an intuitive intelligent assistant interaction framing. We envision that such an interface would not supplant voice, but complement it. Users who wish to maintain the intelligent assistant framing can use this messaging interface upon entering an unfamiliar smart space when voice would be difficult to use successfully. After learning about the space's capabilities and what language the assistant understands, the user would be empowered to switch to voice or another interface if so desired. We see messaging as an addition to the ecosystem of interfaces that users can choose between when operating smart spaces, rather than a replacement for any interface. At the end of our study, we ask users about when they would prefer to use text over voice, as well as voice over text, to better understand the unique advantages of each modality.

5.1.1 The Smart Room Chatroom

There are many possible dynamics for messaging an intelligent smart space assistant. A user could message an intelligent assistant that is associated with them, like a personal assistant who is always the same no matter what smart room the user is in. Alternatively, the intelligent assistant could be tied to a smart room instead of to a user, and the assistant could appear as a messaging contact when the user enters the room.

A third potentially interesting option is that the intelligent assistant is tied to the room, but messaging takes place in a group chat that is also tied to the room—a kind of "smart room chatroom." The communal nature of such a room-level chat log could lead in a number of directions. The chat history could serve as an easy way for users to retrospect on how their decisions affect others in the space, or attribute the current state of the room to the previous decisions of particular people. Furthermore, having a publicly accessible chat log would allow for institutional knowledge about a space to seamlessly and naturally spread in a community, even between members who will never actively communicate with each other. Also, these chat logs could be augmented with additional features, such as the ability to flag issues, make requests, or summon a building manager to the chatroom, which could help facilitate the kinds of human-human interactions that we already know occur during the iterative process of automation rule development and smart space maintenance [24, 27]. A smart room chatroom could also be a quick way to share links, files, and other information with everyone located in the same physical space.

Since in this study we are focused on evaluating different autocomplete designs, our interface is agnostic as to which of these approaches is taken. However, we feel a smart room chatroom has the potential to support the interactions that occupants want to have within and about smart spaces, providing a substantial value to users of these augmented buildings, and is worth further exploration.

5.2 Designing Autocomplete for Discovery

The goals of an autocomplete design for supporting smart space discovery are different than those for traditional autocomplete tasks like search. Autocomplete interfaces are usually designed to help users search a very large, sometimes effectively infinite space of possible results. They are also designed with the assumption that the autocomplete interface will always be available whenever the user wishes to perform a task. Consequently, traditional autocomplete prioritizes the speed and ease of performing search tasks while using the interface, and does not attempt to help users build a complete mental map of the entire space while performing the targeted tasks. The interface does not need to support independence from autocomplete since the user will usually use the same interface to perform future tasks, and in many cases the user cannot build a comprehensive mental map anyway, since the search space is so large.

Smart spaces, however, have a relatively small, finite set of devices and capabilities. Additionally, users can use a number of different interfaces depending on their preferences for each task, including graphical apps, voice input, or text-based input, and even physical buttons and switches. In smart spaces, therefore, autocomplete should be viewed as one possible interface that not only supports performing targeted tasks in a new space, but also bootstraps users' ability to use the other interfaces as soon as possible.

Further, in smart spaces that consist of a mix of smart and dumb devices, it is just as important to be able to quickly determine that you *cannot* control a particular device as

it is to discover that you can. Therefore, in addition to accomplishing tasks and learning about actions that are possible, a discovery mechanism must also help users realize when a particular state is *not* achievable.

To accomplish these goals, autocomplete interfaces must balance performing a specific task (exploitation) with learning about the overall space (exploration). This is a different set of requirements than in traditional autocomplete domains like search, and necessitates the exploration of new designs.

5.2.1 Proposed designs

We propose three dimensions of autocomplete design that we hypothesize will impact the trade-off between accomplishing tasks and learning about the overall space. Each dimension has two mutually exclusive design options. The choices for these three dimensions may interact with each other producing emergent effects, so we evaluate all eight combinations of choices, as well as a ninth baseline interface that has no autocomplete suggestions.

Typing order. In search applications, autocomplete provides suggestions when the user types a whole natural language phrase in-order, but also when the user types keywords, which may appear anywhere in the results [97]. In the smart space domain, should users be given suggestions when they type keywords out-of-order? For example, should typing "projector" allow a user to discover all of the utterances referencing the projector no matter the word's location in the sentence, such as "turn on the projector" and "is the projector on"? While being able to search the set of valid sentences for a substring regardless of its location is useful for rapid search tasks, it may not help users practice the sequential thought process needed to generate utterances end-to-end, as is necessary for the voice modality. It may also stymie learning about the other available capabilities in the space, prolonging dependence on the autocomplete interface in future interactions.

The typing order design dimension describes whether the autocomplete engine will provide suggestions for words typed out of order. For example, if out-of-order typing is allowed it will show the result "turn on the projector" if the user types "projector." If instead in-order typing is enforced, then the user will need to type valid sentences from beginning to end to see suggestions. That is, they must type "turn on" to get the result "turn on the projector." We hypothesize that forcing users to type in order, the way that they would say sentences aloud, may help with retention and exposure to other capabilities. We designate interfaces with these characteristics as either **I (in-order)** or **O (out-of-order)**.

Manual selection of suggestions. Modern autocomplete designs almost always allow users to manually select a suggestion using arrow keys, clicking, or tapping. However, what if users could read but not select suggestions, and had to type them out? Selectability is likely faster and easier, but is essentially a physical gesture. Requiring users to engage their language generation faculties may better entrain users in syntactic patterns and encourage incidental semantic and lexical discovery.

The manual selection design dimension determines whether a user can use up/down arrows, clicks, or tabs to manually indicate an autocomplete suggestion, or whether they

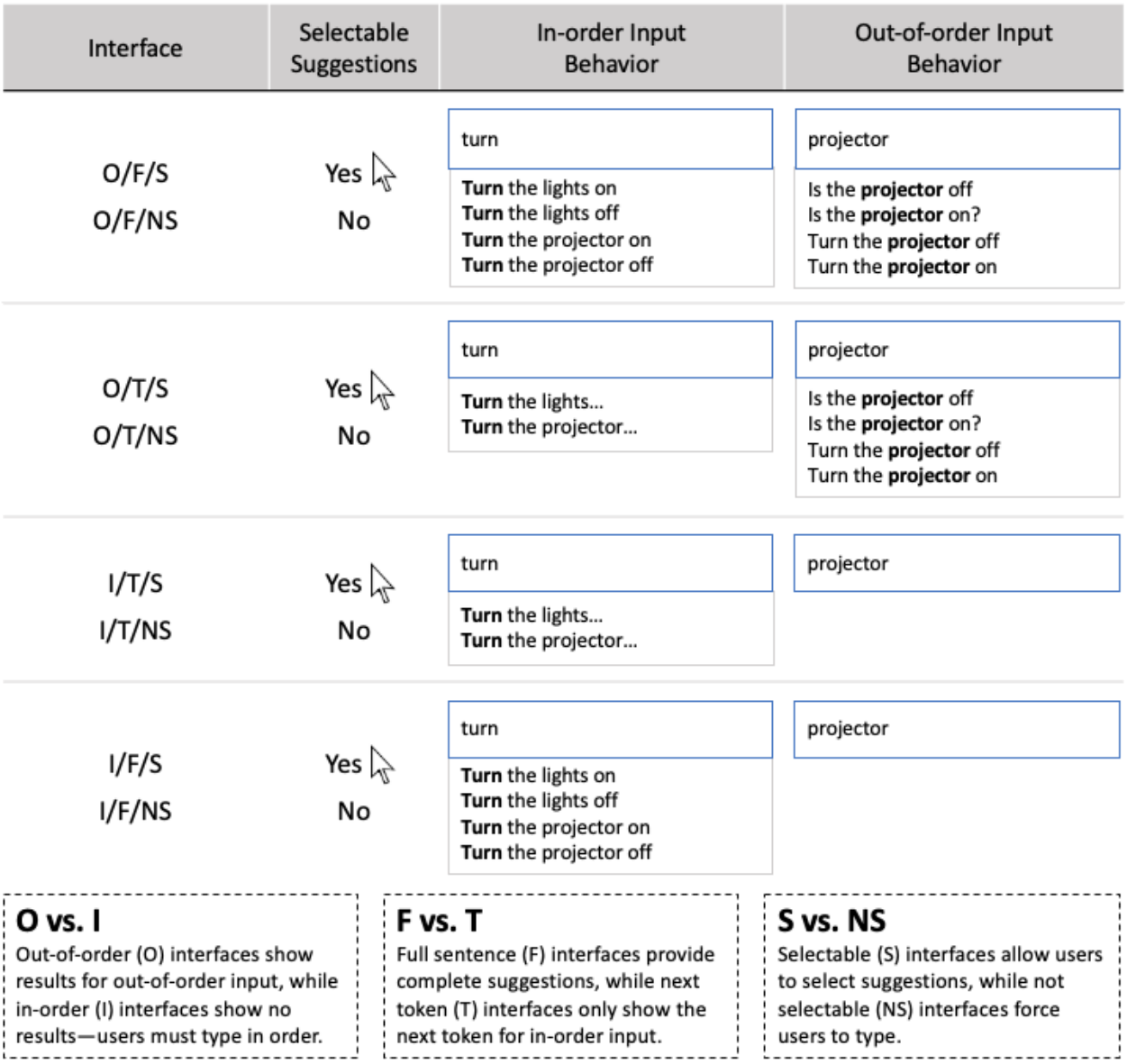


Figure 5.1: The eight autocomplete interfaces and three autocomplete design dimensions. The interfaces are designated by three letters indicating their choices for the three design dimensions. Out-of-order (O) vs. in-order (I) describes autocomplete behavior when users provide out-of-order input (input that does not match the beginning of any suggestion, but does match suggestions somewhere else). Full sentences (F) vs. next token (T) describes autocomplete behavior when users provide in-order input (input that matches the beginning of some suggestions). Finally, selectable (S) vs. not selectable (NS) describes whether or not users can automatically populate the input field by selecting a suggestion via clicking, tapping, or using arrow keys.

must type the whole suggestion themselves. Similar as above, we posit that being able to manually select options without typing will drastically reduce exposure to the rest of the capabilities in the room and impact mental model construction. We suspect that those who use a physical modality to provide input will have worse recall of names afterwards than those who must employ language generation faculties. We designate interfaces with these characteristics as either **S (selectable)** or **NS (not selectable)**.

Extent of suggestions. Autocomplete interfaces in the wild will suggest complete phrases, or just the next word, or both in different contexts [97]. Which is the best for the smart space domain? Are full sentences overwhelming and visually complex? Does limiting suggestions to just the next token lose the opportunity to quickly reveal all the options near the end of the sentence? Does the tree-like search through the next-token expansions result in structured learning?

The extent of suggestions design dimension reflects whether users are shown only the set of next tokens or the set of full sentences. Next token suggestions will only include the next possible words (e.g., if the user types "turn," they will receive the suggestions "turn on," "turn off"). Full sentence suggestions will be complete commands or questions (e.g., if the user types "turn," they will receive the suggestions "turn on the lights," "turn on the projector," "turn off the lights," "turn off the projector," etc.). We hypothesize that those with full sentence suggestions will be more exposed to other capabilities in the room, as they immediately see the devices at the end of the sentences. However, it is possible that next token suggestions encourage users to practice producing input one token at a time and structure the exploration process. We designate interfaces with these characteristics as either **T (next token)** or **F (full sentence)**.

5.3 Autocomplete Study Testbed

To determine how the various autocomplete designs affected the balance between accomplishing tasks and learning about the rest of the space, we built a browser-based conference room simulation and simulated mobile chat interface with a smart space assistant (shown in Figure 5.2). This allowed us to run a large number of experiments online, which was important since we had eight different autocomplete designs to evaluate and therefore needed to collect a lot of data. We also had a ninth baseline interface which has no autocomplete at all. In order to give the users of the no-autocomplete interface a good-faith chance at guessing the right device names, we based the names of the devices off of the guessability study results for the simulated conference room, balancing the popularity of each name with its precision (in other words, its uniqueness to the device). This means that the baseline results approximate a best-case scenario for those with no autocomplete.

During a session, the participant interacted with the intelligent assistant using one of our nine interfaces. We recorded performance metrics while the user performed tasks, and immediately afterwards we collected feedback and tested the completeness and correctness of the mental model the user built while performing the tasks. The experimental apparatus itself

Figure 5.2: The autocomplete study interface.

consists of several interacting components: the simulated conference room, the intelligent assistant, the autocomplete interface, and the logging instrumentation.

5.3.1 Conference room simulation

We created a simulated conference room by overlaying partially transparent frames depicting different device states over a base image, and dynamically revealing or hiding images in response to the user's submissions. The devices and device groups in the simulated conference room are the same as those marked in Figure 4.1 from the guessability study. However, in this study, the recessed wall light was not controllable in order to serve as our unconnected device that is not smart. Additionally, we removed individual control of the blinds, so that they can only be controlled together. The proper names of the devices and device groups were *fluorescent lights*, *front lights*, *spot lights*, *right light*, *left light*, *projector*, and *blinds*. The right light and left light could be controlled individually (as *right light* or *left light*), or together (as *spot lights*). The internal simulation state also included values like *brightness*, *humidity*, and *temperature* in the room so that the assistant could answer questions about the environmental conditions. We only ask participants to perform tasks involving the projector, front lights, and the unconnected wall light, so we are interested in seeing what the participants can learn about the rest of what the space can and cannot do.

5.3.2 Intelligent assistant implementation and grammar

We designed an intelligent assistant chatbot called Scarlett, which served as a mediation layer between human input and the smart space's services and capabilities. Scarlett's design centers around a grammar that defines a core set of smart space functionality, with the vision that the grammar could be modularly composed to reflect any set of available devices. Critically, although the autocomplete suggestions shown were only for these grammar-derived "canonical forms," the assistant could understand arbitrary paraphrases of the suggestions. The assistant's flexible understanding was based off of keyword recognition that mapped the user's input onto one of these canonical forms and identified the relevant entities and attributes. This approach was informed by the work "Building a Semantic Parser Overnight," in which a semantic parser is automatically generated for a new domain by crafting a domain-specific grammar of executable canonical forms, then using crowdsourcing to create a dataset of paraphrases for each canonical form, which is then used to train the parser to predict the likely executable canonical form given a paraphrase [137]. While we did not use a training-based parser, we took inspiration from the work for our general approach of using a spanning set of canonical forms for the autocomplete suggestions and then building an assistant that understands paraphrases of them. The grammar defines the following templates:

- What is the `attribute`?
- Tell me the `attribute`
- Tell me whether the `entity/entities` is/are `state`
- Is/Are the `entity/entities` `state`?
- Raise/Lower the `entities`
- Turn on/off the `entity/entities`

The possible values for `attribute` are *brightness*, *humidity*, and *temperature*, and the possible values for `entity/entities` are the proper device names listed previously in Section 5.3.1. The possible values for `state` depend on the type of device being referenced, and are either *on* and *off*, or *lowered* and *raised*.

Any message that was not a valid command or query received a response of "I don't understand." Valid commands from the user would receive either confirmation messages, such as "okay," or "already [on/off/lowered/raised]" if the device was already in the specified state. The latter is particularly informative when users have incorrectly guessed which name corresponds to which physical device. Scarlett was able to answer questions about the state of the room. We also added the ability to handle some basic conversational interactions unrelated to smart spaces, such as "hello," "please," and "thanks." Typical for current smart space assistants, the assistant did not take initiative, beyond posting a "Hello!" welcome message to the chatroom for the user to see upon first opening the interface.

5.3.3 Autocomplete implementation

The autocomplete suggestions were constructed by using ANTLR to generate a lexer and parser from a context-free grammar specification for the language that the chatbot understands [138]. We used the parser code to construct the set of all valid sentences for our small grammar offline and store them in a database. We used different search techniques to query this database to get the set of autocomplete suggestions for each interface. While generating all valid phrases in advance may not work for larger or infinitely recursive grammars, we did it for our testbed to optimize the search speed so that users did not experience lag in the display of suggestions. We manually created the autocomplete interface widget, including the ability to select autocomplete suggestions with arrow keys or the mouse, which we enabled or disabled depending on whether the interface was selectable (S) or non-selectable (NS).

5.3.4 Logging

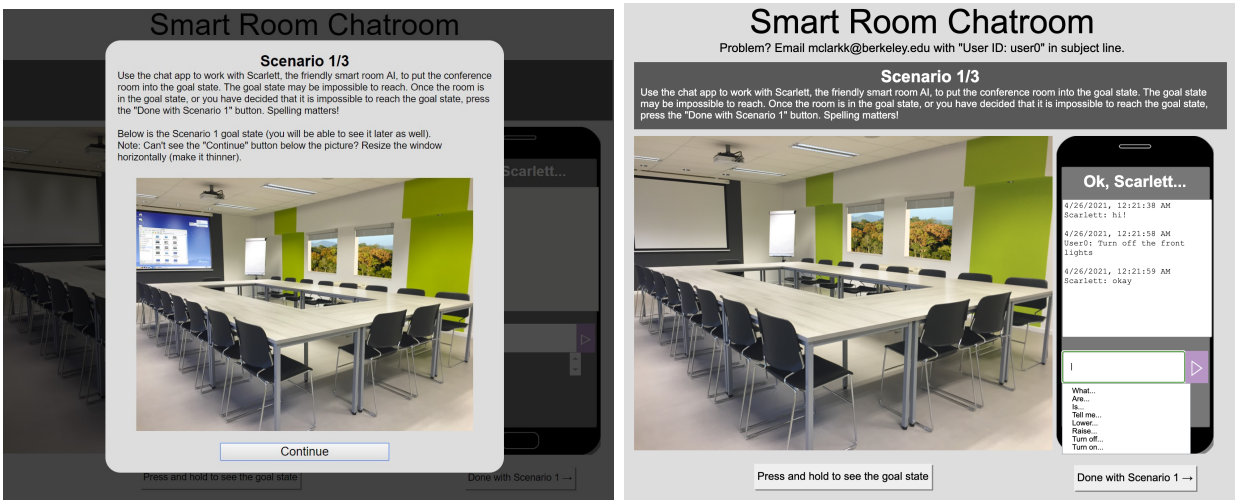
We logged a number of different kinds of interactions between the user and the interface. We logged all keystrokes (including separate statistics for just printable characters), all messages that the user sent to the assistant and the resulting simulation states, and the final state each time the user finished a scenario. We also collected a number of different timing metrics, including overall time, the time spent on each scenario starting from the appearance of the scenario instructions to the end of that scenario, and the time spent just using the interface during each scenario.

5.4 Autocomplete Study Methodology

We posted our study on the effects of autocomplete designs on Amazon Mechanical Turk (MTurk) with IRB approval. We restricted eligibility to the USA to target participants with relevant language skills. Each MTurk task provided a link to our interface's webpage. Upon loading, the webpage received a unique ID from the server to associate with the recorded data. Workers then participated in the experiment, which can be broken down into five phases. Once the worker submitted proof of completion, we compensated them $2.50.

5.4.1 Onboarding Phase

Participants were first brought to an introduction screen which explained the study purpose and what they would be required to do. In addition, users were told that their anonymized data would be shared with other researchers and kept indefinitely, but that they could contact us with their user ID if they wanted to have their data deleted or had any other questions.



(a) Scenario Setup (b) Scenario Interface

Figure 5.3: Scenario workflow. In each scenario, the user is first presented with a setup screen that shows a picture of the goal state for that scenario (Figure 5.3a). Afterwards, the user interacts with Scarlett to put the conference room into the goal state (Figure 5.3b). There are three scenarios in fixed order that build realistically on each other. The goal states for the first two scenarios are each achievable with a single chat message, while the third scenario is impossible.

5.4.2 Interaction Phase

In the interaction phase, the participant was shown a screen with a picture of a desired goal state as well as instructions to place the room in the depicted state, as seen in Figure 5.3a. We used a pictorial approach to avoid biasing participants with our language. The full instruction text is as follows:

> "Use the chat app to work with Scarlett, the friendly smart room AI, to put the conference room into the goal state. The goal state may be impossible to reach. Once the room is in the goal state, or you have decided that it is impossible to reach the goal state, press the "Done with Scenario 1" button. Spelling matters! Below is the Scenario 1 goal state (you will be able to see it later as well)."

Upon clicking continue, the user was presented with the smart conference room simulation and a messaging interface to talk with the smart room's intelligent assistant, Scarlett (Figure 5.3b). The chat history already contained a "Hello!" message from Scarlett. Throughout the scenario, users could review the goal state at any time by pressing and holding a button that changed the simulation to reveal the goal state. Upon releasing the button, the room returned to it current state. Users were assigned to one of the nine autocomplete designs as described in Section 5.2.1 and used the same design throughout the experiment. The user

could use the chatroom to query and command the assistant, with the aid of the autocomplete. When the user decided that they had reached the goal state or that it was not possible to achieve, they clicked a button and a confirmation popup to continue.

The Three Scenarios

During the experiment, users were asked to complete three of these scenarios, each with a unique goal state, with the last scenario being unachievable. The unachievable goal state corresponds to situations where an occupant may want to set the room to a desired state but the relevant device is not controllable or capable of reaching that state. Users should be able to quickly determine the impossibility and move on.

The first scenario involves an easy-to-guess name ("projector"), so that all users, even those assigned the baseline interface without any autocomplete, should be able to achieve the goal. The goal state is for the projector to be on. The second scenario involves a hard-to-guess name ("front lights"). The goal is to remove the glare on the projector screen, so the second goal state shows the front lights above the screen turned off (and the projector still on). The third scenario is to turn on the ambient wall light. However, the wall light is a "dumb" light and is not actually controllable via the interface. At the end of each scenario, the state that the user has placed the room in carries over to the next scenario. Theoretically, the first two goal states should each be achievable with a single message. The ideal outcome for the final scenario would be the users using autocomplete to determine that the light is not any of the names listed, and moving on after making zero attempts.

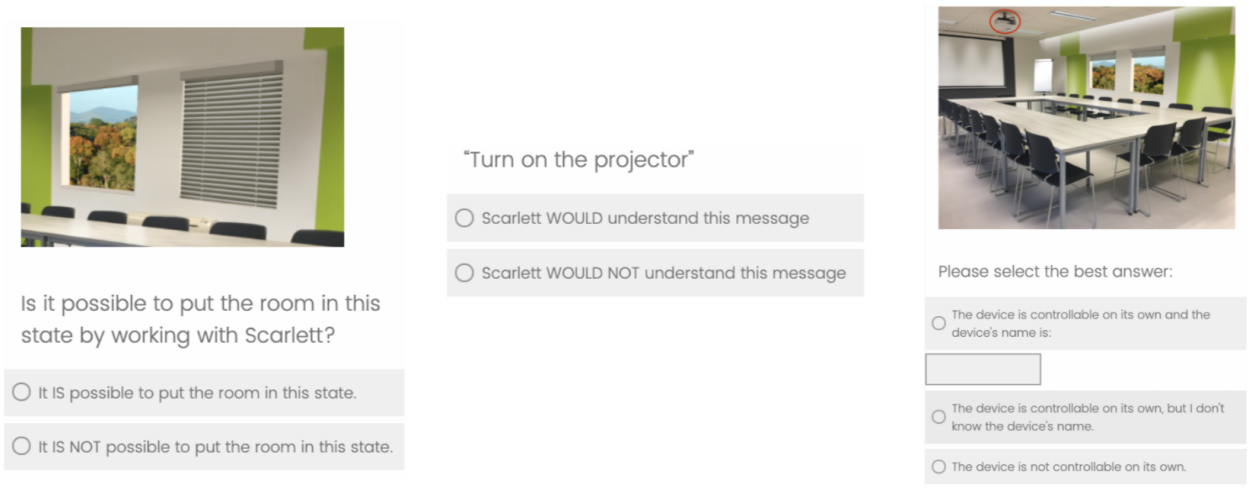
Finishing the third scenario takes users to a final screen that contains a congratulatory message and a link to a questionnaire that they must fill out to receive credit for completing the task. The questionnaire covers the final three phases of the experiment.

5.4.3 Recall Phase

The first phase of the questionnaire is the recall phase. Questions in the recall phase targeted three specific aspects of the users' mental models: an understanding of what goal states were and were not achievable using the intelligent assistant (semantic knowledge, shown in Figure 5.4a), what commands could and could not be understood by the assistant (lexical knowledge, shown in Figure 5.4b), and determining whether various physical devices in the room could be controlled, and if so, what its name was (grounding semantic and lexical knowledge to specific devices in the physical space, as shown in Figure 5.4c).

5.4.4 User Experience Phase

The user experience phase of the questionnaire asked users to share how they felt about the interface. We asked users to report both their enjoyment and frustration with the experience on a Leikert scale, and explain why they provided the ratings they did. We also asked users to list two things that they liked about the smart room chatroom interface, list two things



(a) Semantic (b) Lexical (c) Grounding

Figure 5.4: Recall questions. To evaluate the completeness and accuracy of participants' mental models of the room after using the interface to complete three scenarios, we ask users three types of recall questions in the questionnaire.

they disliked, and describe what features they would add or change and why. We finished the section by asking users to describe at least two situations where they would prefer to use the chatroom instead of voice commands to talk to Scarlett, and two situations where they would prefer to use voice instead of the chatroom. We also administered the ten-question System Usability Scale questionnaire [139].

5.4.5 Demographics Phase

The questionnaire ended with questions assessing participants' level of prior exposure to Internet of Things technologies and intelligent assistants, as well as collecting demographic information such as age group, gender, education, profession, and technical background.

Upon submission the questionnaire displayed a completion code, which the participant submitted on Mechanical Turk as their proof of completion. To prevent learning effects, workers were only able to participate once using one of the interface variants, then they were prohibited though the MTurk platform from accepting the task again.

5.5 Analysis

We ran our study on Mechanical Turk with 210 participants. The lowest number of users for any particular interface was 22, and the highest was 25. Our analysis includes both quantitative and qualitative data. The quantitative results are drawn from the user effort,

accuracy, mental model, and satisfaction metrics. The qualitative results are drawn from participants' free-form feedback on what they liked and disliked, what they would change, and their anticipated preferences for using voice or text chat interfaces.

We want to understand how the different autocomplete designs trade off between performance on individual tasks, learning the overall set of system capabilities (which can be recalled once the interface has been removed), and user satisfaction.

To analyze the interfaces' performance on tasks, we look at *user efficiency*, including the time it took users of each interface to accomplish their tasks, and how many keystrokes and submissions it took them (Figure 5.5). We also look at *user effectiveness*, particularly whether or not the goal state was achieved when the user finished the scenario (Figure 5.7).

To evaluate how many of the overall semantic capabilities and proper names of the system the user learned and could recall afterwards, we analyze the results of the recall phase from the questionnaire. We examine how often users of each interface correctly recognized whether or not a particular state of the room would be achievable by working with the assistant, whether the assistant would understand a particular message, and finally, whether a particular device could be controlled, and if so, what its name is (Figure 5.6).

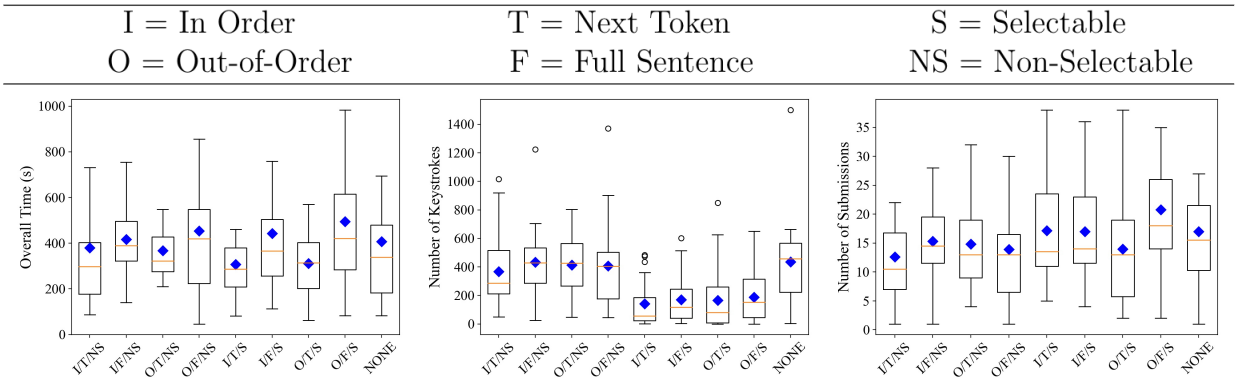
User satisfaction was gauged using a 5-point Likert scale (Figure 5.8) and SUS scores (Figure 5.9), as well as by reviewing the qualitative feedback users provided.

For the quantitative data, typically an ANOVA test would be used to determine statistical significance. However, our data set is non-parametric and unbalanced, which means we cannot use ANOVA. Furthermore, since medians are a poor summary statistic for much of our data, many non-parametric tests, such as a Kruskal-Wallis H test, are also not applicable. Due to these limitations, instead of a statistical significance test we offer box-and-whisker plots of our data set and invite readers to examine the distributions directly.

For readability, we chose not to display the handful of outliers on the plot for time, but information about time outliers is still represented since the outliers were included when calculating the statistics. A few time outliers were so extreme that they distorted the axes to the point where differences that are significant to the application (on the order of tens of minutes to perform smart space tasks) looked visually small. By not showing the time outliers, we can more easily see the general differences between interfaces. The noisiness of the time data is a product of our unsupervised online data collection methodology, as it was possible for users to step away during the experiment to do something else. We reviewed the transcripts and most of the worst time outliers (who had spent nearly an hour on the scenario screens) had perfectly reasonable submissions, so despite their noisy time data we left them in the dataset.

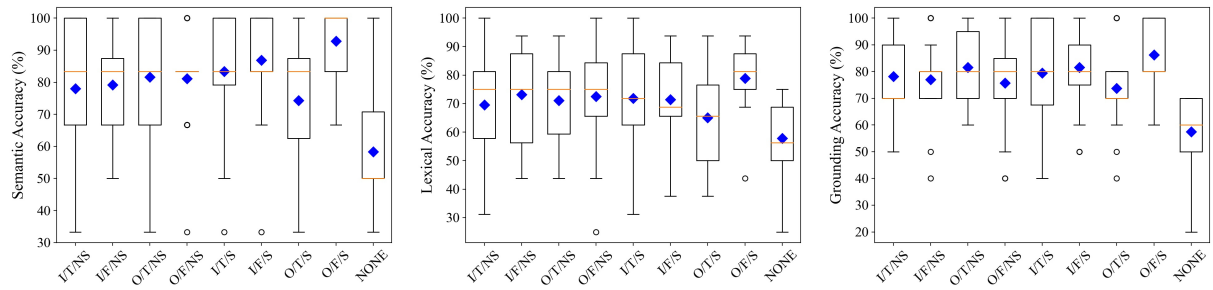
5.6 Findings

We found that while any autocomplete interface is better for accomplishing the task and learning than having no autocomplete at all, within the autocomplete interfaces there were few differences. We also found that users of all interfaces had a difficult time determining



(a) Overall Time (b) Overall Keystrokes (c) Overall Submissions

Figure 5.5: User efficiency metrics for each of the nine autocomplete designs. We approximated user effort by looking at three metrics: (a) overall time in seconds spent on the three scenarios, (b) overall number of keystrokes, and (c) overall number of chat messages submitted to the intelligent assistant. The blue diamonds indicate the means. The time plot (a) does not display outliers, but outliers are included in all mean calculations.



(a) Semantic (b) Lexical (c) Grounding

Figure 5.6: Mental model metrics for each autocomplete design. We evaluated the knowledge users gained about the overall room and interface, not just the components that were involved in the goal states. Each of the three metrics here correspond to the three types of questions illustrated in Figure 5.4. The grounding accuracy numbers only reflect whether or not the participant correctly guessed whether the device or group was controllable and does not consider whether the name was correct.

I = In Order	T = Next Token	S = Selectable
O = Out-of-Order	F = Full Sentence	NS = Non-Selectable

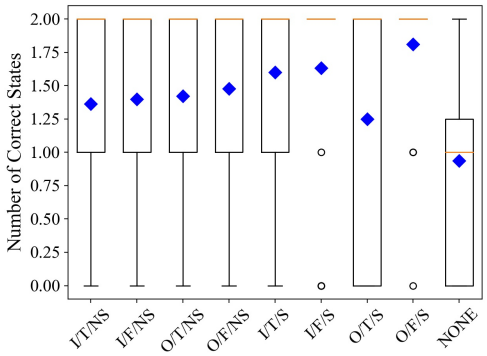


Figure 5.7: Effectiveness in achieving the correct scenario states for each autocomplete design. Each user could obtain up to two correct goal states, since the third goal state was impossible.

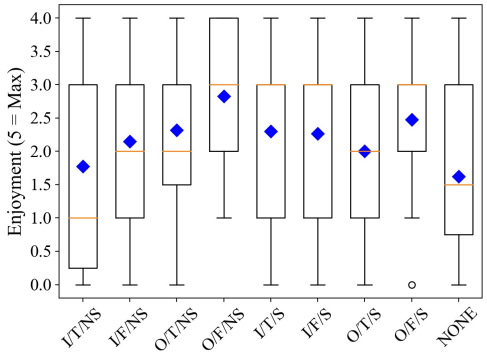


Figure 5.8: User satisfaction for each autocomplete design. After completing all three scenarios, users were asked to rate on a 5-point Likert scale how enjoyable they found using the interface.

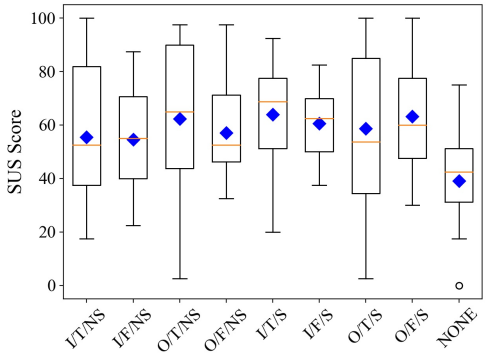


Figure 5.9: System Usability Scale scores by interface.

when a particular device was *not* controllable, which meant they spend a lot of time and effort making fruitless submissions in Scenario 3 instead of quickly moving on. The large number of submissions made in Scenario 3 might explain why there were not many differences between autocomplete interfaces with regards to helping users learn a mental model of the overall space—all interface users presumably learned much of the semantic and lexical information while exhausting the available options. Finally, we also found that users found the text modality useful for more than just discovery, establishing it as a potentially useful smart space interface in its own right, even after the initial discovery phase in an unfamiliar smart space.

5.6.1 Any autocomplete is better than none

The users with no autocomplete, labeled "None" in the result figures, typed and submitted a comparable amount to the autocomplete interfaces (Figure 5.5), but still could not put the simulated room into the correct final state for the hard-to-guess scenario (Figure 5.7), did not learn what the room could do (Figure 5.6), and did not enjoy the experience (Figure 5.8). The interface with no autocomplete performed the worst out of all nine interfaces on the three learning metrics, task completion, and SUS score. This suggests that when users are in an unfamiliar smart space, any autocomplete interface is better than none.

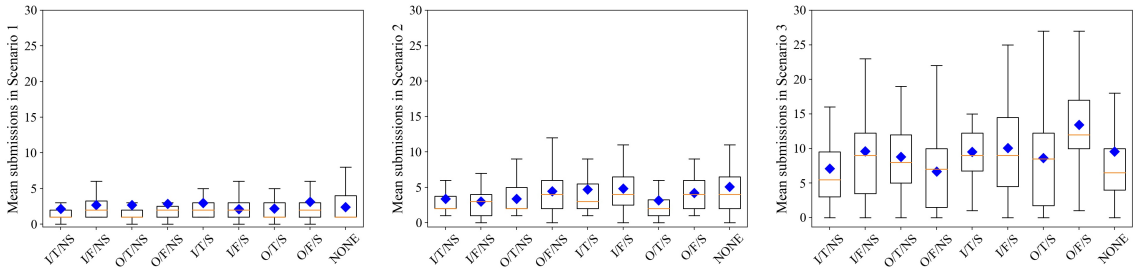
5.6.2 Autocomplete designs showed few differences

The autocomplete designs showed few differences in terms of user efficiency, effectiveness, mental model learning, and SUS score. However, we found a few minor trends.

Figure 5.5a shows that next-token (T) interfaces, which display just the small set of possibilities for the next word as the user types, were always faster than their full-sentence (F) counterparts that display the larger set of entire sentences. Despite the increased speed, the next-token interfaces were not necessarily better-performing or liked more by users, further highlighting the differences between using autocomplete for search tasks vs. smart space discovery—speed is not users' primary concern here. One possible explanation for why the full-sentence interfaces might take longer is that they require more reading.

One surprising result is that the selectable interfaces (S), where users can select the autocomplete suggestions to populate the input box instead of having to type the input themselves, have fewer keystrokes, but did not accomplish the tasks in less time. It appears that instead of using the saved keystrokes to accomplish the tasks faster, they tend to make more submissions.

There was one notably well-performing autocomplete interface, O/F/S, which is the interface that most resembles the autocomplete design commonly used for search. User input can appear anywhere in the suggestions, the results are full sentences, and users can select the suggestions to automatically populate the input field. O/F/S performed the best on the three learning metrics and on effectiveness, and also showed high user satisfaction. However, this may be because O/F/S users made the most submissions, most which were during the



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

Figure 5.10: Submissions by interface for for each scenario. Scenario 1 requires users to turn on the projector, a device with an easy-to-guess name, and indeed users move on after few submissions. Scenario 2 requires users to turn on the front lights, which we know from the guessability study is a hard-to-guess name, so we expect users without autocomplete to take more submissions. However, it is also a high-precision name, meaning if two users use the name, they are likely referring to the same device. Due to the high precision, we thought that autocomplete users would be able to match "front lights" to the correct device, but surprisingly autocomplete users also take a multiple submissions to find the correct name. In Scenario 3, the goal state is not possible to achieve because the device is not connected. We find that all users struggle to determine when a device is not connected, and tend to exhaust the autocomplete options before correctly giving up.

impossible Scenario 3 (Figure 5.10). This means that even though O/F/S initially appears to slightly outperform the other interfaces, that might just be a side effect of the fact that it took O/F/S users many more submissions to determine that the wall light was not a smart light, thus acquiring more experience with the room than other interfaces.

In fact, we suspect that many of these results, and particularly the lack of many significant differences, could be largely due to how extensively all users interacted with the room during their struggles with Scenario 3. We discuss this in the next section.

5.6.3 Grounding names to devices remains a bottleneck

Figure 5.10 is a key figure for understanding the results of this study. It shows how many submissions each interface made for each of the three scenarios. In the first scenario, which required users to guess an easy-to-guess name ("projector"), most users made few suggestions before moving on, even those with no autocomplete, as expected.

In the second scenario, which required users to guess a hard-to-guess name ("front lights"), we would expect users who lack autocomplete to make more submissions, and that is indeed what we see. However, we also know from the guessability study that though it is hard to guess, "front lights" is a high-precision name, meaning that if two people use the name "front lights" to refer to a device, there is a very high chance they both mean the same device. We would therefore expect that autocomplete users, who can essentially see the list of devices

names, would be able to scan the list of names and determine that "front lights" must refer to the lights over the projector screen. However, we see that many autocomplete users also make more submissions, requiring multiple trial-and-error attempts before choosing the correct name. This was confirmed by the feedback from users, who expressed that *"it was tedious to try and find which light was which,"* and *"it was annoying not knowing the label of each device and needing to use trial and error."* Another user reported, *"I got a little frustrated trying to figure out what was called what, but I think once you figure out the names of each item it would be much easier."*

In the third scenario, users were presented with a goal state that was not possible to achieve using the smart space system, as the device depicted in the goal state was not a smart device. This represents situations where users might have a goal in mind, but the system cannot do it. Ideally, autocomplete users would scan the list of device names and realize that none correspond to the target light. Instead, we see that users of the autocomplete interfaces made many submissions. Examining the transcripts reveals that users tended to start with likely guesses, and then after those failed, step systematically through the list of suggestions for each device name. Users usually stopped after trying "Turn on [all] the lights" and seeing that the target light stayed off. This scenario highlights the limitations of autocomplete's ability to quickly provided a mental model of which devices are—and are not—available to operate in the smart space.

The results of the last two scenarios show that while autocomplete helped users learn the list of device names, which physical devices were controllable, and what those devices could do, users were not able to achieve the goal of one-shot actions in unfamiliar smart spaces, as autocomplete was insufficient for helping users map the names to the devices without trial-and-error. The grounding problem thus remains unsolved.

Users proposed various solutions to the grounding problem. Several users suggested that we pick better names for our devices. However, we picked names based on the results of the guessability study in Chapter 4, choosing names that were among the most guessable and also the most unique to each device, making this a best-case scenario. From the results of that study, we know that choosing better names is not a feasible solution.

Users also suggested *"add[ing] a menu [that] describes what each object in the room is named so you can find each item. For example, a diagram with each light in the room and what they are named would have been helpful."* Possible implementations of this approach could be using an augmented reality interface to display device names next to the devices they refer to, or displaying a digital blueprint of the room where devices are labeled either manually or dynamically based on localization system readings (a kind of "Marauder's Map" for devices [140, 141]).

One user suggested that we should support *"clicking on the image to show the commands that control the thing being clicked on. This would save from typing and help with discovery."* This could be implemented in multiple ways, either with a gesture-based or gaze-based interface that allows users to point or look at each item and then receive options. An augmented reality interface would also allow for interacting with devices to see commands. We explore some of these directions in the next chapter.

5.6.4 When text is the preferred modality

Previous studies have touched on the advantages and disadvantages of voice as a modality, especially as compared with graphical user interfaces or text. The hands-free aspect of speech interfaces has been found to benefit users who are simultaneously performing manual tasks, users who have difficulty reading or operating touchscreens, and users who frequently struggle to locate a remote control device [87, 142]. In general, voice has also been shown to heighten the perception of intelligent assistants as a social actor compared to text interfaces [143]. However, when users are sharing highly sensitive information with an assistant, text interfaces are just as effective as voice interfaces at eliciting a perception of the assistant as human-like [144].

In our study, we assume that both voice and text interfaces to an assistant would be available to users, and therefore focus especially on understanding the contextual factors that might impact when a user might prefer one over the other in any given moment. We asked users to describe situations in which they would prefer to use a text-based interface over voice, as well as situations in which they prefer voice over using text. While we believe that the semantic and lexical knowledge gained through autocomplete can also benefit other interface modalities like graphical remote control apps and physical switches, for these questions we focused on the two main natural language modalities.

The single most important factor in a user's preference for one modality over another appeared to be the presence of other occupants and the established social norms for interacting with them. The vast majority of users stated that they would prefer to use text when sharing the space with others. Multi-occupant rooms experience periods *"when the room is noisy or when the room must stay quiet."* In a quiet multi-occupant room, or one in which attention is being given to a speaker, users prefer to use text to avoid disruption. Some example situations that respondents gave were when people are working, taking tests, giving presentations, holding meetings, having phone calls, watching films, and trying to get infants to sleep. In a loud multi-occupant room, users also prefer to use text because a voice interface will struggle to hear them over the noise. Additionally, seven users expressed a preference for text because they feel self-conscious either when speaking in front of others or when speaking to an empty room. Further emphasizing the influence that other occupants have on users' decisions, four respondents said that they would prefer to use voice when alone in a space, and four more mentioned that they would use voice when they wanted *"to show off to others."* One respondent expressed that she would use voice over a text-based interface *"when I'm trying to set an example to my students and don't want to look like I'm texting."*

Unsurprisingly, physical ability also played an important role in modality preference. Ten users mentioned preferring text in contexts in which typing is more physically comfortable than speech. These conditions were usually temporary, as in the case of sore throats or eating, but one user did share that they had a chronically soft voice. For similar reasons, people preferred to use voice over text when their hands were occupied or injured.

The relative cost in time and effort also played a factor. Users strongly preferred voice in situations where they were busy or in a hurry and wanted something done quickly. Even

when not in a hurry, there were contexts where users expected that it would be too much effort to open a chat window, such as when their phone was far away, or when typing a successful command would take more time than just saying it.

Five respondents expressed appreciation for the discovery aspect of autocomplete. Users mentioned using text specifically when unfamiliar with a space: *"I would [...] prefer to use a smart chat room when I am using a room for the first time, so I could see the list of available functions."* One user expressed a preference for text with autocomplete in all situations, saying, *"I'm not sure that I would prefer a voice interface at all. Maybe I would prefer it if it gave me options for what to say, but I think the chat room was easy and straightforward to use, especially because it listed options."* Another user reported that they would use text for discovery, but switch to voice as soon as they had built a mental model: *"If I am using a room over and over again, and am familiar with the functions available, a voice interface would be faster and easier to use."*

Unexpectedly, six people also mentioned that they would prefer to use text when controlling a room remotely. Perhaps this is due to an assumption that remote control would take place while users are out in public and they would like the privacy of text. However, another likely explanation is that users strongly associate voice with embodied intelligence, based on past experiences with the voice interfaces currently on the market and media portrayals of smart spaces. Voice interfaces would therefore be considered natural to use when physically present in the space, but text messaging would be more natural when away.

Taken holistically, these responses confirm that the text-based modality is a communication medium in its own right. Factors besides aiding discovery may lead to users preferring text-based interactions in certain contexts, particularly in multi-occupant spaces, which is worth exploring further.

5.7 Summary

In this chapter, we proposed using autocomplete to support semantic and lexical discovery for smart spaces. However, the unique characteristics of the smart space domain compared to classical autocomplete domains like search necessitate the development of a new set of metrics and revisiting the autocomplete design space. Not only must autocomplete designs for smart spaces aid users in accomplishing targeted tasks, but they must also help users learn overall information about the space that can help them transition to other interface modalities, such as voice, graphical remote control apps, and even physical interfaces, as quickly as possible.

We proposed three autocomplete design axes that might affect the trade off between task performance, learning, and user satisfaction. We evaluated eight different autocomplete designs exploring the combinations of choices for each of these three design axes, as well as a ninth baseline interface with no autocomplete. Users used these interfaces to manipulate a simulated conference room using a "smart room chatroom" interface that allowed them to message the room's intelligent assistant.

We found that all of the autocomplete designs outperformed the no-autocomplete baseline on helping users accomplish the tasks and learn about the room, leading to greater user satisfaction. When it comes to messaging intelligent assistants about unfamiliar smart spaces, it seems any autocomplete is better than none. However, we did not find many differences between the autocomplete designs. This is possibly due to the fact that all interfaces struggled with the final scenario, where they must determine that a particular light is *not* controllable. Instead of quickly moving on, users exhaustively tried all options, potentially wiping out differences that might have emerged after the initial tasks. Were this issue resolved, it is possible that differences might yet be found between autocomplete designs.

More generally, our questionnaire responses indicate that a text modality for smart space interactions could serve an important role in office and home environments, regardless of how well other modalities operate. In addition to the benefits of autocomplete, text is strongly preferred in many multi-occupant situations to avoid disruptions, respect speakers, protect privacy, and avoid self-consciousness, as well as when it is too noisy for voice. Additionally, text is useful for controlling the space when illness affects the ability to speak loudly or comfortably, and could allow for people with speech disabilities to participate in the smart space. Text also appears to be the preferred modality for remote control of a space using an intelligent assistant framing.

Overall, the main takeaway from this study is that while autocomplete helped users learn semantic and lexical knowledge about the smart space in an intelligent assistant framing, users still needed to use trial-and-error to match names to the correct devices, and users struggled to determine when a capability was *not* supported, a critical task in spaces where only some devices are controllable. In other words, the grounding problem remains to be solved.

Chapter 6

One-Shot Interactions with Augmented Reality Photo Messages

Content in this chapter is in submission as Clark, et al. "ARticulate: One-Shot interactions with intelligent assistants in unfamiliar smart spaces using augmented reality." IMWUT.

In the last chapter we introduced the notion of borrowing a messaging metaphor for communicating with agents, and used autocomplete to reveal what was available in the room. However, we found that simply providing users with a list of names is not sufficient for achieving one-shot interactions in unfamiliar spaces. Names on the list could apply to multiple devices in the room—what Furnas referred to as "the precision problem." This manifested as two main issues: 1) Users took multiple attempts to guess the correct name for smart devices, and 2) users struggled to tell when an appliance was not a smart device, instead trying and eliminating all the available names before determining the device must not be connected.

To solve this remaining problem, we need to support grounding discovery—a way to help users map the available devices' names to the specific devices that they refer to. Additionally, we want to do so within an interaction framing of communicating with an intelligent assistant.

In this chapter, we explore using augmented reality to label the devices with their names. We maintain the intelligent assistant context by extending our previous messaging approach from a text-only "chatroom" metaphor to a photo-and-text "Snapchat" metaphor [145]. Snapchat is a popular smartphone app for social messaging. The app opens into a live camera view that allows users to snap a quick photo of their immediate surroundings, caption it, and send it to a contact. Inspired by this workflow, we design a messaging app called ARticulate, where the live camera view reveals smart devices and their names using augmented reality. ARticulate users can take a picture of the devices, caption the image with the aid of autocomplete suggestions that are tailored to the devices in the image, and then send the captioned image to the smart space intelligent assistant.

In this chapter, we evaluate the ARticulate design by running user studies with nine participants. We find that while users of the baseline voice interface struggle to operate an unfamiliar smart space, ARticulate users are able to achieve the goal of one-shot interactions

in unfamiliar smart spaces, as well as zero attempted interactions with unconnected devices. Additionally, we find that the knowledge learned during use of this messaging interface is able to translate to the voice interface afterwards, unlocking a previously inaccessible part of the smart space ecosystem after only a few interactions. By successfully designing a smart space interface that prioritizes the ability to easily operate unfamiliar smart spaces, we greatly improve the usability of existing systems and remove a significant obstacle to widespread adoption of smart space technologies.

6.1 Design Goals

The design goals for our system are similar to the design goals for the autocomplete interface, though our methods for achieving these goals is different. An interface design that helps users independently discover and use the capabilities of an unfamiliar smart space should achieve the following four design goals:

Goal 1: One-shot interactions with connected appliances in unfamiliar smart spaces. Users should be able to walk into an unfamiliar smart space, discover what functionality is available, and invoke the desired functionality in the first attempt.

Goal 2: No attempted interactions with unconnected appliances. As a part of discovering what functionality is available, users should also learn what functionality *is not* available. Users should be able to determine when a device is not a smart device without making any failed interaction attempts.

Goal 3: An assistant-oriented interaction framing. As with the autocomplete interface, while we want to enable one-shot interactions in unfamiliar smart spaces, we want to do it specifically while preserving an agent-oriented interaction metaphor. This is because we found in Chapter 3 that assistant-mediated interaction framings help users imagine valuable high-level applications of smart space technologies, and thus will be a key interface for the long-term future of smart spaces. Intelligent assistants also currently provide fewer clues about what devices are available and how to invoke them compared to smartphone apps that display device icons, making assistants the smart space interface in greatest need of discovery support. Centering the design on intelligent assistants means that the user should feel as though they are interacting primarily with the assistant, not the devices.

Goal 4: Learning transferable information that enables the use of other smart space interfaces. If the user could build a broad map of the overall capabilities of the space, beyond those involved in their specific tasks, the knowledge would unlock the ability to use other interfaces for future tasks, even if those interfaces lack discovery aids. Since most smart space systems are designed around proper device names, this means helping users learn what smart devices are available in the space and what their proper names are, even if they are not relevant to the current task. This broad knowledge could help enable use of interfaces such as voice that do not support discovery but have other contextual advantages (such as being hands-free).

If we are successful, we should see that users can use the app to do possible things in one shot, and learn that impossible things cannot be done without having to suffer through failed interactions. We should also see users clearly struggle with voice in the beginning without the app, but that if they use the app it enables them to successfully use voice afterwards. We should also see that users feel they are messaging the assistant rather than directly manipulating devices. We evaluate our design on all of these goals.

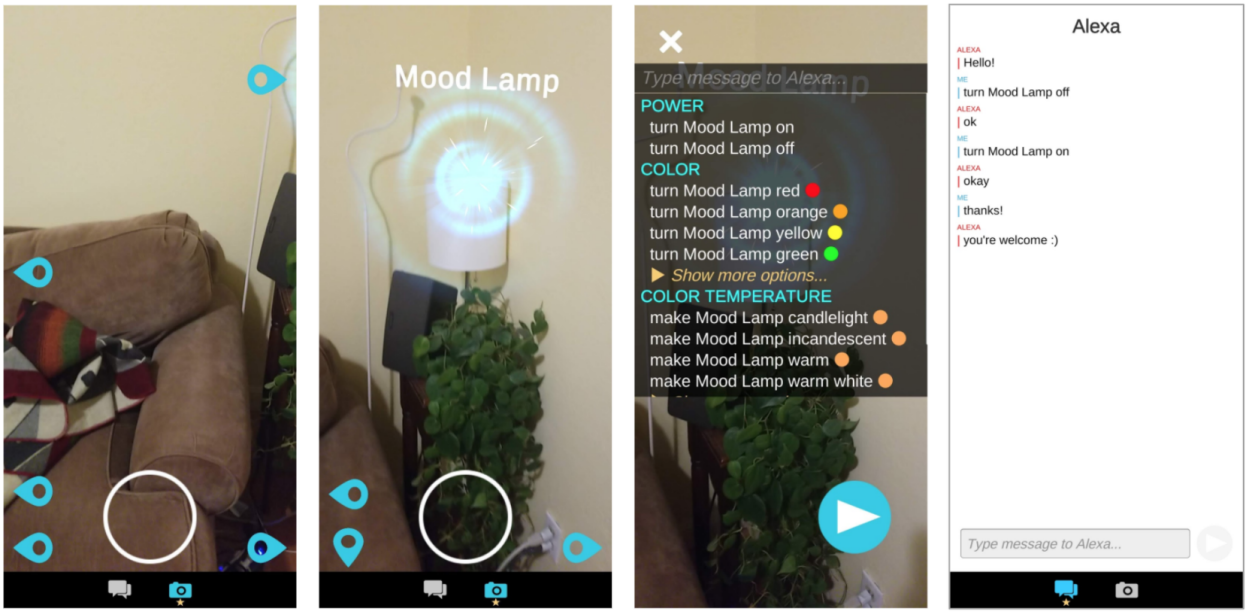
6.2 Augmented Reality Messaging for Smart Space Assistants

We designed an app called ARticulate, a messaging app for communicating with smart space assistants that uses augmented reality and autocomplete mechanisms to support discovery. In our vision, there is one assistant for a smart room, and while the user may be able to communicate with that assistant through other means, such as a smart speaker, they also have the choice to message that same assistant with the ARticulate app. The ARticulate app uses augmented reality labels in a camera view to show users which devices in the environment are smart and what their names are. The user is able to snap a picture of a device they want to control and caption it with a command for the assistant, aided by autocomplete. That photo message is then sent to the assistant for execution.

6.2.1 Ephemeral photo messaging for contextual communication

To draw on users' prior experiences with messaging and ensure our app design had street-tested usability, we wanted to model the communication workflow with the assistant on technology that people already use to communicate with each other. We chose to model ARticulate's workflow after Snapchat, a popular messaging app with 293 million daily active users worldwide [146]. In the United States, 65% of adults aged 18-29 use Snapchat [147]. Snapchat's main usage centers around photo messages, which are pictures of the users' immediate environment with an optional caption, sent to specific contacts. Snapchat users are familiar with the incorporation of augmented reality into photo messages, as the app provides "lenses" that transform the face, and "world lenses" that embed virtual objects in the environment [148].

Snapchat opens directly onto a live camera view, encouraging users to snap a quick photo of their immediate environment, caption it, and send it to one or more friends. The central conceit of Snapchat is that by default the pictures disappear after viewing, which the designers intended to encourage a culture of more frequent and informal photos compared to curated permanent photo collections [149]. In practice, users do primarily use Snapchat for a style of messaging where photos are valued for their situational relevance [116–119]. Out of all social media, this characteristic makes Snapchat an archetype of photo-centric messaging that puts a strong emphasis on the immediate physical context of the sender.



(a) (b) (c) (d)

Figure 6.1: ARticulate interface. ARticulate is a smartphone app for communicating with smart space assistants inspired by the Snapchat messaging application, with key modifications to support smart home discovery. Off-screen markers that slide around the edge of the screen indicate the presence of nearby smart devices (a). Glowing animated orbs with labels indicate the location and proper names of on-screen smart devices (b). Users can take a picture and caption it with a message for the assistant. Autocomplete suggestions (c) aid in caption composition, providing insight into what the agent understands and what capabilities the device has. Suggestions are included for all on-screen devices. A chat history window (d) allows users to see a record of what captions were sent and the assistant's responses, and also provides users with a way to type directly to the assistant without the necessity of photos.

6.2.2 The ARticulate user experience

While ARticulate models the overall messaging workflow on Snapchat, drawing upon its well-tested interface design and its familiarity to many users to help users quickly conceptualize sending contextual messages to a smart space assistant, the augmented reality and autocomplete feedback represent significant innovations that support smart space discovery.

As shown in Figure 6.1, ARticulate allows the user to use the live camera view to look around the smart room and discover the location and names of smart devices (as well as which devices are not smart by their lack of a label). Off-screen indicators point in the direction of the various smart devices in the room, and when the device is in view of the camera, the fact that it is a connected device is indicated by a glowing orb and its name placed on its real-world location. The user can use the shutter button to take a picture at any time, and then caption the picture with a message to send to the smart home assistant. Captioning is

aided by autocomplete suggestions spanning the set of available actions that the assistant understands about the devices in the photo. When the picture is sent, the user is taken to the chat history window, which shows the sent messages as well as the assistant's responses.

While technically the assistant does not need the photo itself to correctly perform the tasks in our study, the interaction metaphor of "sending" the assistant a picture of the physical context along with the message is useful for framing the communication about the user's immediate context. This lays the groundwork for future interactions where the user's intent is expressed in a combination of information in both the photo and the caption. For example, while ARticulate's autocomplete suggestions for captions are unambiguous standalone phrases that will also work with smart speakers, our implementation also allows users to omit device names from captions and rely solely on the photo context for scope. Concretely, users can snap a photo, caption it with "red," and any device in the photo that supports color settings will be set to red. We did not evaluate this deictic interaction mode in our study and our users did not discover it, but it illustrates the potential role of photos in communication. Sending intelligent assistants photos with captions could potentially become a staple mode of communicating with assistants about context-specific topics.

In ARticulate, the user is communicating only with the smart space assistant, rather than fellow humans; the captions are aided by autocomplete suggestions that inform the user what the assistant can understand; and there are augmented reality markers showing the locations and names of the various smart devices in the room. The first change is based on the interaction framing work in Chapter 3, the second change is from our explorations in Chapter 5 on using autocomplete to teach users what language the assistant understands, and the final change solves the precision problem and turns trial-and-error into one-shot interactions. Together, these unique features enable a short yet informative smart space discovery process, presented interactively as part of an interface for communicating with an intelligent assistant.

6.3 Implementing the ARticulate Augmented Reality Messaging App

To evaluate whether this design works for our stated goals, we built an Android app that implements all the key features described above. The Android app also made some other minor design choices that while not critical for success, may nevertheless affect overall usability of apps of this nature, so we discuss them below. Additionally, some of the implementation choices are important to discuss even though they are not fundamental to the system design because they broke users' expectations or otherwise affected user experience during the study. We also built a smart home testbed for the ARticulate app to interact with, and a chatbot for users to message. We discuss the details of our implementation in the next sections.

6.3.1 "Alexa" chatbot

In designing ARticulate, we envisioned that there would be one smart space assistant for the entire room, which the user could choose to either talk to through, for example, a smart speaker, or message through an app like ARticulate that contains additional support for discovery. In our implementation, we chose to use Alexa as the underlying smart space assistant, as Alexa-enabled devices such as the Amazon Echo and Echo Dot are among the most popular smart speakers at this time [21]. While Alexa itself could potentially be improved for smart home tasks, helping users discover how to successfully interact with such a popular interface as it is, *especially* given any challenging idiosyncrasies, would illustrate the effectiveness of the ARticulate design for supporting discoverability. However, Alexa does not provide an API that allows developers to send text-based utterances. This means that we had to preserve the illusion that the assistant the user communicates with over chat is the same Alexa as the Alexa that the user can speak to using the smart speaker.

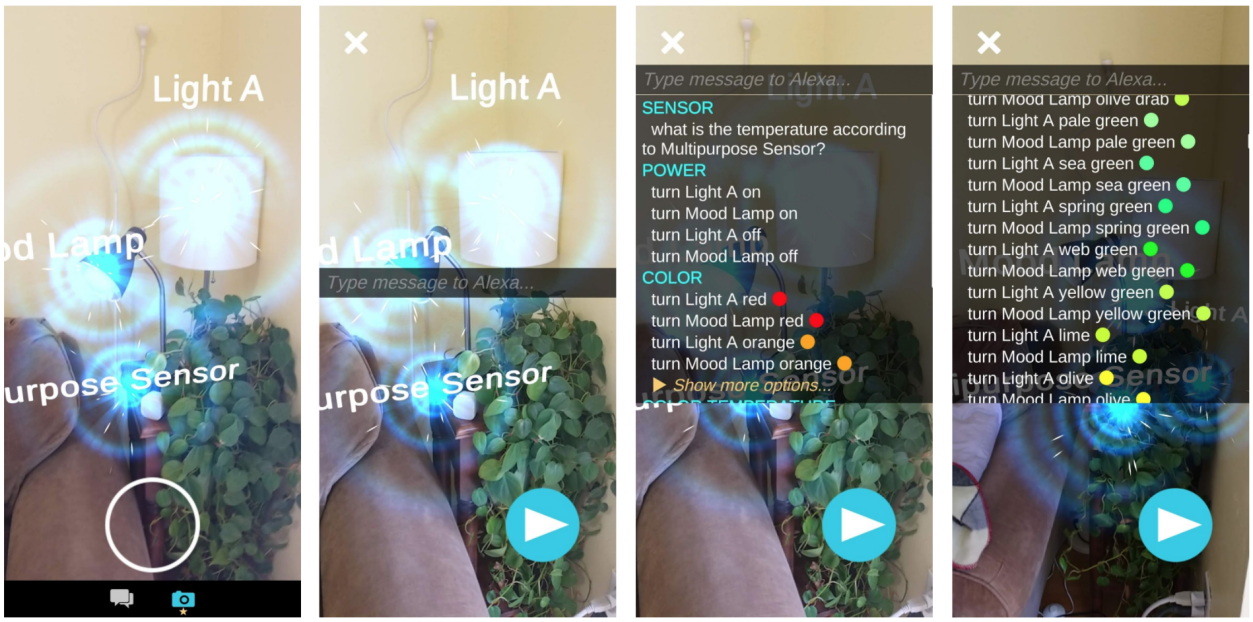
We wrote a backend server that implemented a chatbot that closely imitated Alexa's responses and behavior. Manipulation of smart devices was done through cloud access to the smart home platform. However, this chatbot was not a perfect clone. Alexa can understand and respond to non-smart home interactions, such as answering questions by looking up information online, which we did not attempt to duplicate. We also did not duplicate functionality like setting timers, playing music, or responding to queries about the weather. These choices were intentional as we were focused on use of the devices in our smart home testbed. However, we made one key error in our duplication of smart device functionality – our chatbot could not understand commands involving "all the lights." This was a problem in one instance, when Participant 4 noticed that the text-based "Alexa" in the app could not understand the command "Turn all lights pink" and "Turn on all lights," while the voice-based Alexa could, and was quite upset about the inconsistency. We clarified during the debriefing that this was an oversight and that in an ideal implementation of the system it should be the same assistant whether messaging or speaking out loud.

6.3.2 Autocomplete

Because the app should be able to work in any smart space with its own assistant, the app did not come with any suggestions pre-programmed. Instead, it received the set of autocomplete suggestions from the backend chatbot server upon connecting. To make the suggestions render faster, the server sent the entire set of possible suggestions up-front, and the app filtered them locally depending on what the user photographed and typed.

Due to the computational overhead of providing autocomplete suggestions for every possible paraphrase that the system understands, the autocomplete suggestions were only provided for "canonical" phrasings like "Turn Mood Lamp on," and "Turn Floor Lamp red," even though the chatbot also understood many paraphrases of each command.

As shown in Figure 6.1 and Figure 6.2, the autocomplete suggestions showed entire suggestions as full sentences rather than just the set of next possible words. Suggestions



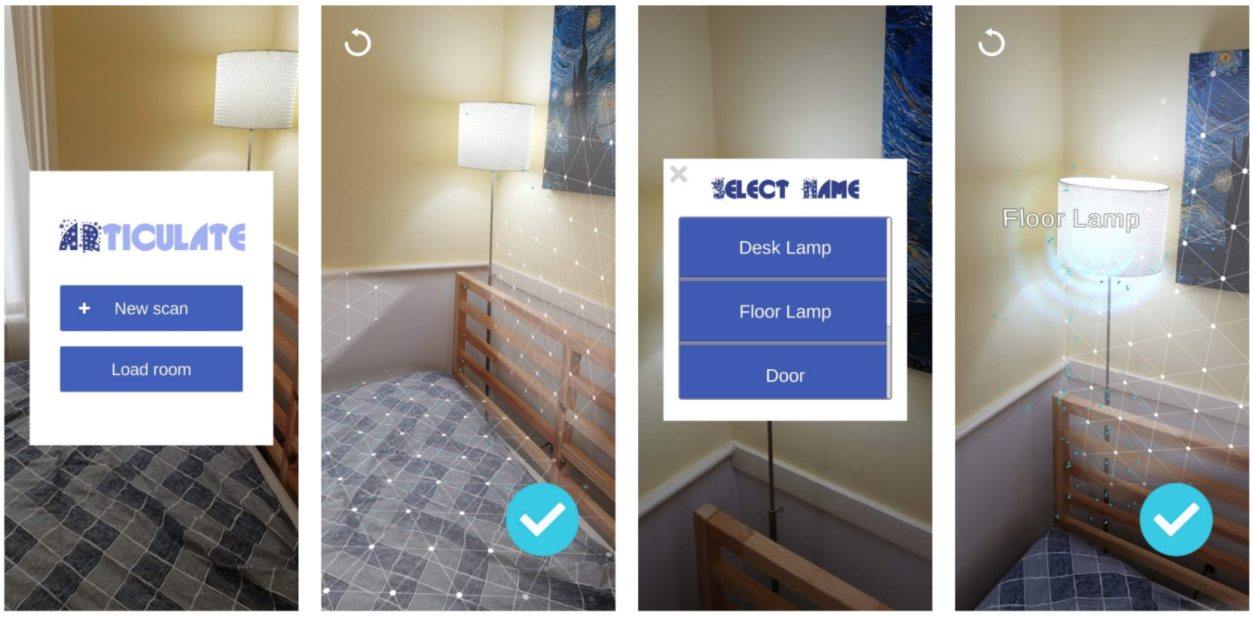
(a) (b) (c) (d)

Figure 6.2: Autocomplete suggestions for multiple devices. ARticulate users can view (a) and photograph (b) multiple smart devices at once. In this example there are three devices: Mood Lamp, Light A, and Multipurpose Sensor. The autocomplete design has several features to help structure the large number of resulting suggestions (c). Suggestions are categorized into "capabilities" like *power* and *color*. When multiple devices are in a photo, the autocomplete suggestions are interleaved by capability. Each capability section hides excess suggestions with a "Show more options" drop-down. This was especially critical for the color suggestions, as Alexa understands over 145 different color names – too many to casually scroll over. An illustration of the variety of color suggestions that are revealed after tapping "Show more options" is shown in (d). These design choices are intended to help users quickly understand the range of what the available devices can do with minimal scrolling and reading.

would be shown if the input the user typed (e.g. "blue") appeared anywhere in the suggestion. Users could tap on suggestions to place them into the user input box. This corresponds to the out-of-order, full-sentence, selectable (O/F/S) design in Chapter 5.

To help users organize the large number of suggestions into broad categories of available actions, suggestions were divided into sections based on "capability" like *power*, *color*, *color temperature*, *brightness*, *lock*, and *sensor*. These capabilities were derived from the device schemas provided by the SmartThings IoT platform API we used to control the devices, and thus were programmatically generated. When multiple devices were in one photo, the autocomplete suggestions for the different devices were interleaved by capability.

To allow users to quickly scroll over the available types of capabilities and build a broad mental map of available actions, each capability section hid excess suggestions with a "Show



(a) (b) (c) (d)

Figure 6.3: ARticulate scanning process. To prepare for the study, the on-site facilitator used the smartphone to make a new 3D scan of the room (a), which involved moving slowly around the room while the vision system detected planes (b). The facilitator manually placed device markers by tapping the phone screen over a detected plane. After selecting the appropriate label (c), the marker was then placed on the plane (d). This vision-based and heavily manual approach to localizing devices was burdensome for the facilitator and performed poorly under changing lighting conditions, but we expect these challenges to be greatly reduced by upcoming technology improvements.

more options" drop-down. This was especially critical for the color suggestions, for example, as Alexa understands over 145 different color names (the vast majority of which are unique device color settings) – too many to casually scroll over.

6.3.3 Augmented reality

To display the locations and names of smart devices, implementations of this design will need to be able to access and display information about the locations of devices in the space. We envision that in real-world settings, a scan with this information could potentially be made once during the initial system setup as a part of configuring the device, and automatically shared with users who later enter the space.

For ARticulate, we provided a scanning feature that allowed users to create a scan of a smart space stored locally on the phone, illustrated in Figure 6.3. This feature was used exclusively by the study team to configure the app for usability testing. We implemented the augmented reality functionality in Unity using Google's ARCore library. ARCore is an

entirely vision-based method for augmented reality. To create a scan, the user must slowly move the phone camera around the smart room, during which time the ARCore library detects vertical and horizontal planes. The user can then tap the screen to place a device marker. The library determines the placement by raycasting from the tap and finding the point on the plane that intersects with the ray, which is where the device marker will be anchored. After tapping to place a device marker, the app provides a list of possible device names, which the user selects as appropriate to set the label.

This scan-and-place process can be tedious, so we implemented a number of additional processes and app features to help with the scanning process. The vision-based process that is used to detect planes and localize itself within the space is very sensitive to lighting conditions. To deal with this, we used a flat folder with a visually distinct pattern to help aid plane detection in places such as the top of lampshades, turned on all the lights in the room that were not involved in the study to keep the light consistent throughout the session, and closed blinds and window shades. In a couple instances, we waited until nightfall because we found that scans made during dusk's changing lighting conditions would stop working soon after completion. We also scanned the room from as many angles as possible to ensure the app would recognize the room regardless of where the user went. The app had the ability to undo device placement an arbitrary number of times in reverse order, allowing the user to redo placement, potentially for all the devices. We discuss the challenges involved in scanning more in Chapter 7 when we discuss the future of these systems.

6.4 Evaluating ARticulate in Unfamiliar Spaces

To evaluate whether ARticulate achieves the four design goals we laid out in Section 6.1, we would like to observe users performing tasks in an unfamiliar smart space with a mix of connected and unconnected devices. As a baseline, we would like to see how hard it is to accomplish the tasks using state-of-the-art voice assistant interfaces, as well as whether the ARticulate design improves the ability of users to accomplish the tasks and reduces the number of failed interaction attempts. We would also like to see whether use of ARticulate enables successful use of the voice interface afterwards, including for tasks that the user has not done before. To answer these questions, we devised an IRB-approved study protocol where we sent a smart home testbed directly to users' homes and asked them to perform tasks using both voice and ARticulate.

6.4.1 COVID-19 pandemic challenges

Designing the protocol for evaluating ARticulate was unusually challenging due to the COVID-19 global pandemic. During this time, most citizens in the study country (the United States) were under some form of shelter-in-place order that prohibited them from leaving their homes except for essential reasons, and research facilities and university campuses were closed. This placed major constraints on the study protocol design.

Since our research goal is to test how well our system supports user discovery in an unfamiliar smart space, ideally we would bring users into an on-campus smart space that we set up and that is unfamliar to them. However, this indoor gathering of strangers was not permitted under the COVID-19 lockdown. Instead, we made a portable testbed and sent the testbed to each participant. However, the participants needed an on-site facilitator who could set up the testbed for them, so that they could approach the system with no knowledge. To ensure that there was an on-site person already in the participant's "germ bubble" who could set up the system, we added our entire research group to the official IRB study team and had every member undergo human subjects research training. The recruitment pool was therefore limited to people who lived or were otherwise already within contagious contact of our lab members. This is a form of snowball sampling, which is subject to biases that we will discuss during the overview of our participants in Section 6.5. This approach also imposed additional requirements onto the design of the testbed system to be easy to install, configure, and operate remotely.

6.4.2 Recruitment

The principal audience for this test were people who might need to operate an unfamiliar smart space. Examples include people likely to be guests in a smart home, inhabitants of a smart home managed by someone else, lodgers in a smart hotel room, workers in a smart office space, and visitors in a smart conference room. Since this potentially includes many adults, we defined our recruiting criteria as people over 18, with a home internet connection with WiFi, and fluency in the English language. As mentioned above, due to COVID-19 restrictions, recruitment used a form of snowball sampling that drew from residents and close contacts of members of the study team.

6.4.3 Smart home testbed

To evaluate ARticulate's performance in an unfamiliar smart space, we sent an entirely self-contained set of testbed devices to participants' homes, ensuring that while the space itself was familiar to the users, the smart system setup was novel. This familiarity with the normal room and novelty of the testbed devices does pose an issue for evaluating discovery, however. It would be reasonable for users to assume that every new fixture or device added to the room for the experiment has smart capabilities, while every appliance that was previously present is not part of the smart room. If these assumptions were true, it would give them an unrealistic advantage in evaluating whether or not a device is smart.

To counteract this, one of the fixtures we added is a lamp that is not connected to the smart space at all. Also, one of the lights we included, Light A, was installed in an existing fixture that is normally in the room and is normally not smart. In this way, we were able to assure that whether or not a device is normally in the room provided no additional information about whether or not it was smart.

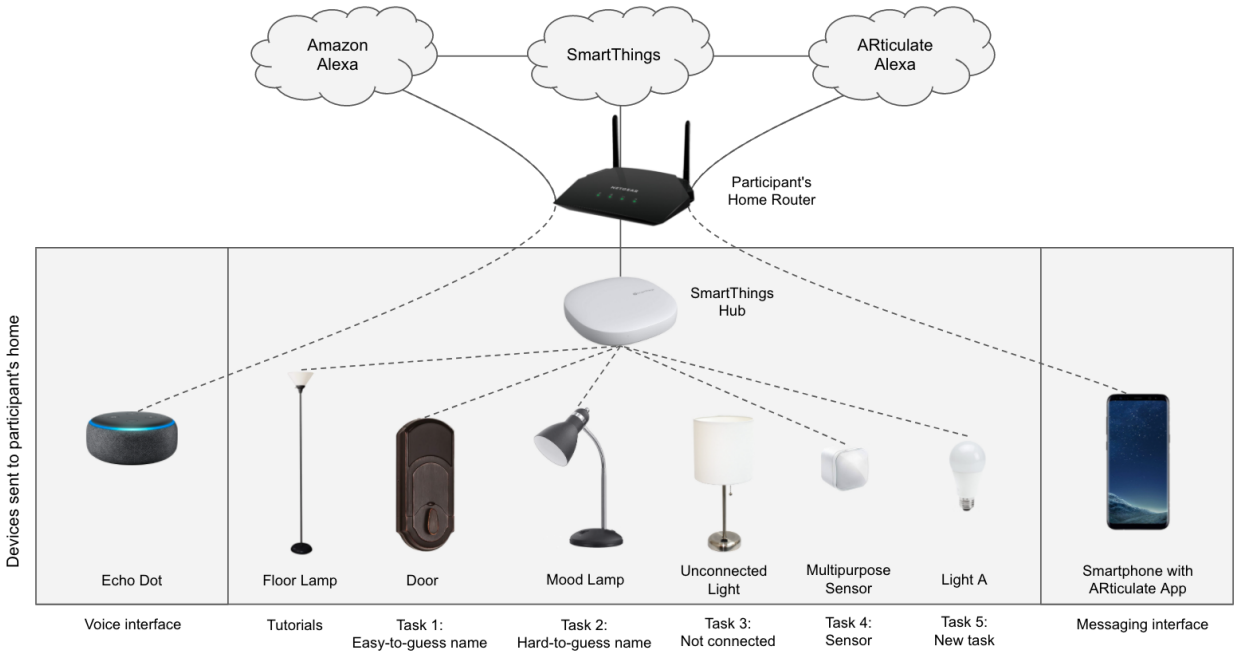


Figure 6.4: ARticulate testbed. To evaluate ARticulate during the COVID-19 pandemic, we created a testbed kit that we could send to each participant's home. Each participant lived with a study team member who could set up the testbed and facilitate the in-person needs of the study, such as running the video call and using a laser pointer during task instructions to indicate relevant devices to the participant without using language. The smart home testbed itself consisted of six devices, each of which corresponded to a different task that tested some aspect of the research question. The "smart" appliances connected wirelessly to the commercial SmartThings platform [150] through a hub. Each participant used two different interfaces to communicate with the assistant at different phases of the study: a voice interface via the Amazon Echo Dot [151], and the ARticulate messaging app preloaded on a smartphone. The cloud servers powering these interfaces interpreted the participant's utterances and actuated or queried the smart devices through the SmartThings API.

The complete configuration of the testbed is shown in Figure 6.4. Devices in our smart space testbed consisted of three smart LED bulbs that all supported power, color, brightness, and white temperature, one smart door lock that could be remotely locked and unlocked, and an environmental sensor that measured temperature, motion, and light (though Alexa only exposes temperature). We connected all of these devices to the internet through the SmartThings hub [150]. The devices communicated with the SmartThings cloud platform, which acted as a rendezvous point for any service that wanted to interact with the devices, such as Alexa or ARticulate.

There were two main advantages to using SmartThings. The first is that the testbed could be easily moved from location to location because the system setup just required connecting the SmartThings hub into the WiFi router of the new site via Ethernet. This meant at each new site, we could provide network connectivity to the entire system simply by plugging one device into the router, with no onboarding process or passwords required. The second advantage is that having a single platform allowed us to easily set the system to a known state before the interactive user sessions by tapping a single button in the SmartThings phone app to activate a custom saved scene.

We chose this set of smart devices to correspond with particular tasks, each of which spoke to a particular research question.

- Floor Lamp. The floor lamp, fixture included, was a device we included for tutorial purposes, since we did brief tutorials for both the voice interface and ARticulate.

- Door. The door lock was the device with the easiest-to-guess name. This was an assumption based on the fact that there was only one lock, so we (correctly) expected users to default to "unlock the door." We installed the door lock on a free-standing metal door reinforcement plate, which allowed us to hang the door on an existing doorknob or place it on a surface, such as a floor or a table. This device provided a baseline for comparison against hard-to-guess device names. We expected users to be able to operate this device correctly using either voice or the messaging app.

- Mood Lamp. Mood Lamp, fixture included, was a device with a hard-to-guess name, as most people might call it a desk lamp or small lamp or table lamp, though Mood Lamp is nevertheless a reasonable name as well. We expected this device to be difficult to figure out how to operate using the voice interface compared to using ARticulate.

- Unconnected light. This light did not have a name, since it was just a normal light bulb. This represented situations where a user is in a smart space where only some devices are smart. The user may want to perform a task that is in fact not possible because the device they want to operate is not a smart device. During our study, users are informed that some of the tasks we will ask them to do may be impossible, and they must figure out whether that is the case. We expected this to be extremely difficult to determine using the voice interface, because the voice interface has no way to help users distinguish between cases where the device they want to control is not smart and

cases where the user simply has not guessed the device's name yet. We expect this task to be easy with ARticulate, due to the absence of an AR marker indicating connected capabilities.

- Multipurpose Sensor. This sensor measures temperature, motion, and light, though Alexa only exposes temperature. We included the environmental sensor to explore interactions with sensors through assistants, branching out from actuation. According to the operation classifications we developed in Chapter 3, the lights and door will elicit "immediate actions," while the sensor will likely elicit different operation classes, "direct questions" and "indirect questions." Additionally, the sensor poses a new kind of discovery challenge. Lights and door locks, smart or not, have an obvious presence that makes the fact that there is some affordance or functionality highly visible, even if the user is not quite sure how to use it. Small wireless sensors, however, are intentionally unobtrusive, and their affordances are not visually obvious. Even determining that a small white cube is a sensor at all is tricky. We expect that if not explicitly pointed out to participants, participants using voice will not even be aware of a sensor in the room, much less think to access the data it provides, whereas with ARticulate we expect users to be able to discover and access the data.

- Light A. This final light bulb did not come with a fixture – instead, the on-site facilitator was instructed to install it in a fixture in the room that is normally present. This was to overcome the bias towards assuming that only the new fixtures were smart devices. We did not ask users to perform any tasks with Light A using the ARticulate app. We used Light A to evaluate whether after using ARticulate, a user could use voice to interact with a device that they had not performed a task on before.

6.4.4 Test preparation

In advance of each user evaluation, we used a no-contact method to deliver to the participant's home the evaluation kit containing the smart home testbed, an Echo Dot with Alexa, a smartphone with the ARticulate app, and various paperwork. This delivery was received by the on-site study team member. Over a video call, we worked with the on-site facilitator to set up the smart home system and perform the augmented reality scan of the room, including device marker placement. We held an informed consent discussion of the study where we reviewed the goals and overall structure of the session with the potential participant, and if they chose to sign the consent form, we continued with the rest of the session.

6.4.5 Overview of session structure

The overall session took anywhere from an hour to two hours, depending on the session type participants were assigned. Figure 6.5 illustrates the main components of the study protocol. First, all sessions began with a short pre-interview where we asked the participants about their prior experiences with related technologies that might inform how they approached the

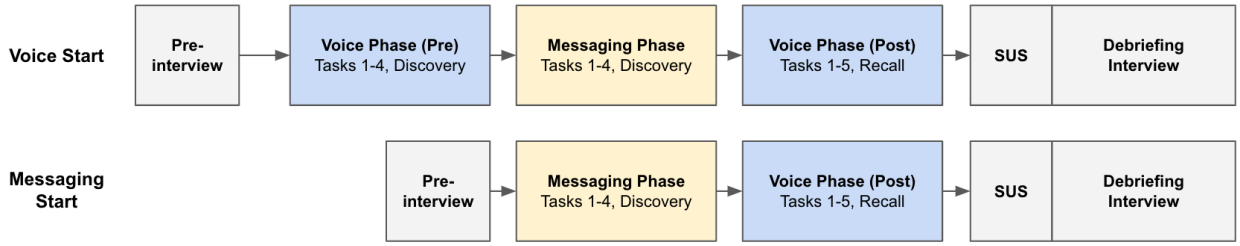


Figure 6.5: User study protocol. Participants were assigned one of two possible session types, Voice Start or Messaging Start. The session type determined whether the user started the interactive tasks using voice or ARticulate. "Voice Start" participants experienced all three interaction phases starting with a voice phase. The "Messaging Start" treatment skips the first voice phase entirely to show that performance during the messaging phase is not due to learning effects. In both the first voice phase and the messaging phase, the tasks are the same (Tasks 1-4), and are followed by a three-minute self-directed discovery period. However, note that the final voice phase contains an extra task (Task 5: New Task) that is not posed anywhere else. The final voice phase also ends with users recalling what they have learned about the devices in the room, rather than discovery. All sessions begin with a pre-interview about user experiences with prior technologies, and end with an System Usability Scale (SUS) questionnaire and debriefing interview.

systems in the study. Then we began the interactive tasks, which were broken up into two or three phases depending on the type of session. After the interactions, the participant filled out a System Usability Scale (SUS) questionnaire [139] about their experiences, which we used to guide a final debriefing interview where we asked the participant detailed questions about their experience and thoughts.

6.4.6 Pre-interview

All sessions began with a short interview about the user's prior experience with related technologies, to understand any biases that might influence users' interactions. The main topics covered were intelligent assistants (including smart speaker usage like Echo and Google Home, but also usage outside of a smart home context, such as on mobile devices), smart rooms (including those controlled by tablets or phone apps), and the Snapchat messaging application. This interview usually took only a few minutes.

6.4.7 Interactive Phase Structure

There were two different session types that participants could be assigned that determined what the interactive portion of the session looked like. The interactive portion was broken up into two or three phrases depending on session type, with each phase specifying which

interface the participant used. Figure 6.5 outlines the general structure of the two different session types. The two session types were as follows:

- **Voice start**. In this session type, the participant went through three phases, first a voice-only phase where the participants were asked to perform several specific tasks and to try to discover capabilities of the room using only a voice interface to Alexa. This phase established a baseline of how difficult the tasks are to complete using state-of-the-art voice interfaces to smart home assistants. In the second phase, the participants were introduced to the ARticulate messaging app and performed the same directed tasks and free-form discovery, but with ARticulate instead of voice. In the third phase, the participants were returned to the voice interface to see if their ability to use the voice interface to perform tasks had improved after using ARticulate. There was also one participant assigned a **Voice start, no sensor** variation. This was the same as the normal voice start, except that during the first voice phase the sensor task was omitted. This was to see whether the participant would notice the sensor and attempt to interact with it using voice during free-form discovery.
- **Messaging start**. In this session type, participants were introduced immediately to the ARticulate messaging app. This was to rule out any learning effects from earlier phases on the performance of the app. After this first phase, users completed the final phase, where they interacted with the Alexa voice interface to see if they were able to successfully use the voice interface more effectively than those who started with the voice interface.

6.4.8 Interactive Tasks

In each phase, we asked users to perform the same four or five tasks using the interface for that phase. In these tasks, the users were asked to interact with certain devices. However, due to the linguistic nature of the tasks, we wanted to avoid priming users with any language about the device itself. The on-site facilitator would state the task instructions, such as "get Alexa to turn on this device for you. Please narrate your thoughts out loud as you do so," while using a laser pointer (included in the test kit) to indicate which device they were referring to. The interactive tasks were as follows:

- Task 1: Easy-to-guess name. For this task, the instructions were to "get Alexa to unlock this device for you." This could be done with the command "unlock the door." Unlocking the door also required a numeric code, which we provided users with in advance and do not count against interactions, since we are merely interested in whether the user can guess the easy name. This task provided a baseline, as we expected users to be able to operate this device correctly in one guess regardless of interface.
- Task 2: Hard-to-guess name. For this task, users needed to turn on the Mood Lamp. Mood Lamp is a device with a hard-to-guess name, as most people might call this a

desk lamp or small lamp or table lamp. We would expect this task to be difficult using the voice interface compared to using ARticulate.

- Task 3: Not connected. In this task, the user was asked to turn on an unconnected lamp. Since this light is just a normal light bulb, the task is not possible. This represents situations where a user is in a smart space where only some devices are smart. The user may want to perform a task that is in fact not possible because the device they want to operate is not connected. Throughout our study, users were informed that some of the tasks we will ask them to do may be impossible, and that they must figure out whether that is the case. Before each phase we reminded users that they could either complete the task or decide it was impossible and move on. We expected this to be extremely difficult to determine using the voice interface, because there is no way for users to distinguish between cases where the device is not connected and cases where the user simply has not guessed the device name. We expected this task to be easy with ARticulate, since the unconnected light lacked an AR marker indicating connected capabilities.

- Task 4: Sensor. In this task, the facilitator instructed, "There is a device in this room that measures something. Get Alexa to tell you what the current value is." This could be accomplished by asking Alexa "What is the temperature according to the Multipurpose Sensor?" We expected this task to be quite difficult with voice, as small wireless sensors are unobtrusive by design, and their affordances are not visually obvious, making the need for discovery aids particularly acute.

- Task 5: New task. This final task was only performed in the final voice phase, where participants had to use voice without any discovery aid except what they remembered learning from ARticulate. In this task we asked them to turn on Light A, which is a device that we had never previously asked the participant to interact with in other tasks. We used this task to evaluate whether after using ARticulate, a user could use voice to interact with a device that they had not interacted with before.

In addition to the tasks, at the end of the first voice phase and the messaging phase, users were given three minutes for self-directed discovery where their goal was to learn as much as possible about the room using the interface for that phase. Additionally, at the end of the final voice phase, there was a recall section where we asked users to share as many device names and capabilities that they remembered learning about the room.

6.4.9 SUS questionnaire and debriefing interview

The System Usability Scale (SUS) is a commonly used usability instrument that asks users to answer 10 standard questions on the Likert scale related to usability issues. Questions alternate between 5 and 1 being the most positive answer for the system to counteract any impulse to fill in all ones or all fives. The total SUS score is out of 100. We instructed users

Treatment	Participant ID	Gender	Age	Assistants (Voice)	Assistants (Text)	Smart Speaker Use	Smart Home Use	Unfamiliar Smart Spaces	Snapchat
Voice start	1	F	41	Yes	No	Yes	Yes	Yes, hotels	No
	2	F	31	Infrequently	No	Infrequently	Yes	No	Yes
	3	M	24	Yes	No	Yes	Yes	No	Yes
	4	F	26	Yes	No	Observer	Observer	No	No
Voice start, no sensor	5	M	21	Yes	No	Yes	Yes	No	Yes
Messaging start	6	F	24	Yes	No	Yes	Yes	No	No
	7	F	28	Yes	No	Yes	Yes	No	Yes
	8	F	28	Yes	No	Observer	Observer	No	Yes
	9	F	28	Infrequently	No	No	Observer	No	Yes

Figure 6.6: Participant demographics and background. The columns show participant demographics as well as whether the participant has used various technologies. The "Assistants (Voice)" column includes both smartphone and smart speaker use, so it is a superset of the "Smart Speaker Use" column. "Observer" is used to indicate when someone does not use a technology themselves, but frequently observes another use it, such as a roommate, partner, or close friend. Strikingly, though speaking out loud to assistants was very common, no one remembered typing messages to an assistant, though both of the main smartphone assistants (Google Assistant and Siri) support it. Also, while all of the participants frequently visited or lived in a home with smart space technologies, almost no participants had been in an unfamiliar smart space set up by someone else that they needed to figure out how to use.

that the SUS questions only applied to the ARticulate interface, not to the smart home itself. We primarily used the SUS as a prompt to guide user thinking through the post-interaction debriefing interview. In addition to going over the user's SUS questionnaire answers and discussing why they answered the way they did, we also asked users questions like "Under what circumstances would you use this? Why?" and "If you had three wishes to make this better for you, what would they be?" and "Which features of the app did you find the most helpful while accomplishing the tasks?"

6.5 Participant overview

We recruited nine participants. Four participants went through all three phases (the "Voice Start" treatment), one went through all three phases but was not given the sensor task in the first voice phase ("Voice Start, No Sensor"), and four skipped the first voice phase and started immediately with ARticulate ("Messaging Start"). The demographics and background experience with related technologies of the participants are shown in Figure 6.6.

The majority of study participants were women. This is largely the result of our snowball recruiting method that enrolled co-habitants, particularly partners, from our mostly male lab. Past studies have shown that while most current smart home "maintainers" responsible for installing, setting up, and debugging the system are men, many of the *users* of the resulting smart home are women [22–27]. These women must discover and use the available functionality, sometimes while they are on their own in the house, even though they did

not set up the system. This means at least in the short term, women would be the ones most likely to use and benefit from a discovery-oriented interface like ARticulate, so our participants resemble a likely user population. In the long term, however, an approach like ARticulate should be useful to any user in any situation where they want to interact with an assistant in an unfamiliar space, including office buildings and public spaces.

Though most participants were not technical, all participants lived in or had been in smart homes before, and the majority had used Alexa or Google Home to operate a smart home using voice, or closely observed others who did. This is likely a result of the snowball recruitment method imposed by the COVID-19 restrictions and may not be reflective of the general population. However, since we are interested in how our system would work in a future where smart spaces are more ubiquitous, it was useful that our participants were already familiar with the domain and did not need to learn about smart homes while also trying to help us evaluate ARticulate.

Despite the fact that most participants had interacted with smart homes before, almost no one had been in a situation where they needed to figure out how to operate an unfamiliar smart room on their own. Consistent with prior work, most participants were in smart homes set up by friends or family from whom they learned how to operate the system, either through instruction or observation [22]. No one had been in a smart office, conference room, or public space, much less one they had to operate primarily through an assistant.

6.6 Findings

We analyzed whether or not users were able to accomplish tasks and the number of interaction attempts each user made for each task. The results are shown in Figure 6.7. The interaction counts reveal that, as expected, users struggled operating the unfamiliar smart space using voice. However, when using ARticulate, in the majority of cases users accomplished their goals in a single interaction, and made zero attempts to interact with the unconnected device. We also found that some of the knowledge learned through ARticulate was able to transfer to voice, allowing users the option of using the hands-free modality after only a few tasks with ARticulate.

6.6.1 Voice users require many attempts to accomplish goals in an unfamiliar space

Users who began the session with the voice interface to Alexa found the tasks difficult. Figure 6.7 shows that, as expected, participants were able to perform the easy-to-guess Task 1 (Door) in one attempt, but for the remaining tasks they were not. For Task 2 with the hard-to-guess device name (Mood Lamp), three of the five participants were able to eventually determine that the device was Mood Lamp and achieve the goal. At first it may seem surprising that they could guess the name. However, this is because Alexa helps the users with prompts, such as “Did you mean Mood Lamp?” (the right answer) in response to

			Participant ID								
Interface	**Task name**	**Ideal**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Voice	T1: Easy-to-guess name (Door)	**1**	1	1	1	1	1				
	T2: Hard-to-guess name (Mood Lamp)	**1**	1	4	4	6†	3*				
	T3: Not connected	**0**	28	8	4	2	3				
	T4: Sensor (Multipurpose Sensor)	**1**	9*	1†	4*	4*					
Messaging App	T1: Easy-to-guess name (Door)	**1**	1	1	1	1	1	1	1	1	1
	T2: Hard-to-guess name (Mood Lamp)	**1**	1	1	1	1	1	1	1	2*	1
	T3: Not connected	**0**	0	0	0	1	0	0	1	0	0
	T4: Sensor (Multipurpose Sensor)	**1**	1	1	1	1	1	1	2	1	1
Voice (Post-App)	T1: Easy-to-guess name (Door)	**1**	1	1	1	1	1	1	1	1	1
	T2: Hard-to-guess name (Mood Lamp)	**1**	1	1	1	1	1	1	1	1	1
	T3: Not connected	**0**	1	0	0	0	0	0	0	3	1
	T5: New Task (Light A)	**1**	1‡	1‡	1	1‡	1	1	1	1‡	1*
	T4: Sensor (Multipurpose Sensor)	**1**	2†	1†	3*	4*	3†	1*	3	1	2*

* Did not accomplish a task that was possible
† Accomplished task, but without using name
‡ Accomplished task, but with incorrect name

Figure 6.7: Number of interactions per task for each participant. An interaction is a message or utterance directed to Alexa.

"Turn on the desk lamp" (a wrong guess). Participant 1 used this feature during Task 3 later, explaining, "It's not the reading lamp, but I'm going to try the reading lamp just in case she gives me the name of this device." However, sometimes Alexa would suggest the wrong light. Remarkably, users would occasionally reject her correct suggestions if they did not think it was the light they were trying to operate, trying to guess the name themselves. This resulted in a wide variety of outcomes in the voice phase, usually with participants making a number of attempts. Participant 4 was eventually able to technically achieve the Task 2 goal of turning on Mood Lamp by asking Alexa to turn on all the lights.

For Task 3 (not connected), the lamp we asked participants to turn on was a normal lightbulb not connected to the smart system. With the voice interface, there is no way for participants to easily determine whether the task is impossible or whether they just have not guessed the right name yet. Consequently, users made many attempts to turn on a light that could not be turned on, with the highest count belonging to Participant 1, who made 28 attempts before giving up. Three out of the five participants gave up after asking Alexa to "turn on all the lights" and discovering that the unconnected lamp did not turn on.

In real-world conditions, it is likely that users would give up and use another method rather than make so many attempts. Participant 4 said, "I think in real life if I were in an AirBnB, I would not have asked again," and also noted, "I think what I would do if I needed to turn on this lamp would be to turn it on myself."

In fact, we observed that several participants physically reached for the device for closer inspection after several failed guesses. Four out of five participants touched or asked to touch the physical fixture, either to look for clues or because they thought the device might be broken. Participant 4 said, "The other thing that I'm thinking about is that maybe there are physical signs that something is a smart thing. So I'm going to look at this floor lamp [the tutorial device] now. [...] Can I take this lightbulb out?" During Task 2 with Mood Lamp, Participant 5 toggled the power switch on the Mood Lamp to make sure it was properly connected after failing to guess the name multiple times. Participant 5 was not the only user who thought a device was broken. During Task 3 with the unconnected light, Participant 1 said: "So there's something wrong with your lamp, you guys, it's like broken or something, or it has a very special name that's not lamp, and it's not light."

The sensing task was even more difficult for voice than the actuation tasks. As Participant 1 summarized, "So, I would have to identify what this device is, then figure out it's name, and then use that name for the measurement reading." Further into the task, Participant 4 said, "I'm personally at a loss, because there are so many different ways to measure something, and not knowing what she's measuring makes it hard for me to ask, and she also can't tell me what she's measuring."

In real-world situations, users may not even realize there is a sensor to try to read. Participant 2 said, "Until the app showed the glowing dot [in Phase 2], I had no idea that [Multipurpose Sensor] was even there." We omitted the sensor task entirely in Participant 5's voice phase in order to see whether he would attempt to interact with it or ask Alexa about sensors during self-directed discovery with the voice interface. He did not. During the debriefing, he reported noticing the small white cube when he entered the room, but it did not occur to him to try to interact.

Helping users discover the presence of sensors, which ARticulate was designed to do, may not only help highlight available functionality, but also may help users with privacy concerns. After hearing the sensor task instructions, Participant 4 shared, "My first thought is that I'm freaked out, because I don't like being measured." The voice interface did not expose the presence of the sensor. Having the ability to immediately locate and identify sensors in a room using an app like ARticulate may help empower occupants by informing them of potential privacy issues.

Though the sensor task was already inherently difficult for voice users due to the fact that sensors are unobtrusive and users must guess what they measure, the task was further complicated by limitations in Alexa's natural language understanding for sensor interactions. The phrases Alexa requires to get sensor readings were complex, and some variations would work, while others would not. For example, "What is the temperature according to Multipurpose Sensor?" works, but "Tell me the temperature according to Multipurpose Sensor" does not work, much to the surprise and frustration of Participant 4, who thought she had remembered it exactly during the final post-app voice phase. Alexa also understands some interactions that omit the sensor name, but not others. "What is the temperature inside?" does work, but "What is the temperature in the room?" does not, though multiple participants tried it. Thus, even though it was possible to get the sensor reading without knowing the device's

name, users were still often unable to guess the correct phrases when using voice.

Alexa's narrow and inconsistent understanding of sensor commands also caused problems for our development process. While writing our chatbot clone, we tried to guess the phrases Alexa understood to get the sensor readings, but only discovered "What is the temperature according to Multipurpose Sensor?" As a result, that is the only autocomplete suggestion ARticulate provides for the sensor. We did not know that there were any valid options at all that omitted the sensor's name until Participant 2 guessed "What is the temperature inside?" during the first voice session. This issue with the complexity and specificity of the phrases, on top of the unobtrusive and opaque nature of sensors, made the sensor task difficult overall.

A remarkably common strategy when users struggled throughout the first voice phase was to ask Alexa for a list of connected devices. Four out of five participants requested a list of devices, usually multiple times. Participant 4 was particularly distressed by the end of the voice phase: "I wish I could somehow ask to understand what all the devices are that she has because I think that would help me [...] I'm frustrated because I want this list!" During self-directed discovery in the voice phase, Participant 3 made eight separate attempts to ask Alexa for a list or count of the connected devices. Surprisingly, considering it is such a common strategy for users, Alexa does not support providing a verbal list of connected devices. Nevertheless, we know from the work in Chapter 4 and Chapter 5 that just providing a list of connected device names without grounding them to their locations would not be a sufficient solution – many proper names could plausibly refer to multiple different devices, and more importantly, a list of smart device names does not help users quickly discover when an appliance they want to control is *not* smart and connected. A list of device names would therefore simply bound the number of trial-and-error attempts, rather than help users truly achieve one-shot interactions.

6.6.2 ARticulate enables one-shot interactions and identifying unconnected devices in unfamiliar spaces

With the ARticulate messaging app as an interface to Alexa, the tasks were easy. Almost all participants accomplished the possible tasks in one attempt. The one notable exception was Participant 8 in Task 2, who was not able to figure out how to turn on the Mood Lamp. This is because she took a photo of Mood Lamp that also had Light A in the upper right corner, but the label for Light A was washed out by light. Even though the photo said "Mood Lamp" in big letters over the target device, and the suggestions interleaved "Mood Lamp" and "Light A," the Light A suggestion was the first one in the list so the user selected it. When it did not work, she then tried typing the same Light A command directly into the chat, which also did not work, so she decided to move on. She realized during self-directed discovery at the end of the tasks that Light A was another light and that the original was called Mood Lamp. However, she still thought that she had only received Light A suggestions previously: "That's interesting, when I tried to change Mood Lamp before, what was coming up was Light A. [...] I feel like that wasn't happening before, but maybe it was. I guess I had

the wrong light, or you're tricking me."

During the debriefing, after being shown how suggestions for multiple devices are interleaved under the capabilities, Participant 8 expressed an expectation for devices as the main subheadings, rather than capabilities: "Oh, I guess I didn't see a different hierarchy for that. So I guess for me that was a little confusing. [...] I would just assume a different heading with a different spacing under it for like, Lamp A all the way to the left, and then power slightly in, and then Floor Lamp..."

There are a number of changes that could be made to the design to prevent this kind of misunderstanding, such as ensuring that labels cannot be washed out by light on the camera, or changing the method for autocomplete into drill-downs, or allowing users to select how suggestions are grouped. However, despite this singular exception, the overall approach illustrated by ARticulate appears to have worked to support one-shot interactions.

Using ARticulate, participants also recognized right away that the unconnected device was not connected, and largely did not attempt to interact with it at all. We can see that the majority of users had zero interactions for Task 3 using the messaging app. The two participants that made attempts on the unconnected device verbalized that they did not think it was connected. Participant 4 said, "This is not a smart device. I can see now, using my special smart device window that there's nothing I can do to get her to turn that on for me, so I'm not going to try. Well, I guess the other thing I could at least try, just in case, is...now that I have her on text, I wanna see if I can get her to tell me what all the lamps are, to just confirm." She typed "List all lamps in room," and received the standard "I'm sorry, I don't know that one" response, at which point she moved on.

Participant 7 similarly observed, "It doesn't have a thingy for it [...] so I think it's not set up?" However, she took a picture just in case. Unfortunately, the default behavior when there are no devices in the photo is to give suggestions for *all* devices that Alexa knows about. The idea behind this was to give users access to suggestions if they were inconveniently located for a photo. However, this design choice was confusing. Participant 7 typed the suggestion "Turn Light A on" since she had not interacted with Light A yet. When a different light turned on, she concluded "I kinda think this one is just not set up to work with this device." This misunderstanding could be avoided just by not showing suggestions by default when no devices are in the photo.

The scores from the System Usability Scale questionnaire were good overall, with a mean score of 83 and a median of 85. The worst-scoring question on average was Question 1, "I think that I would like to use this system frequently." It makes sense that most participants did not think they would use ARticulate often, since as their backgrounds show, they have not needed to operate unfamiliar smart spaces on their own. When asked when they would use something like ARticulate, participants did identify situations when they were in an unfamiliar smart space, they just did not expect that to be a frequent situation. At current time, this is true, but at least partially this is due to a chicken and egg problem—smart space technologies cannot become ubiquitous if they are not casually usable by strangers. As Participant 4 said, "I feel like once I knew the name of everything I would probably prefer to use voice personally, but [...] if I had this system, it would be the absolute first thing I did in

any room that was a smart room. [...] every time I had a new room I would definitely use it."

Participants expressed appreciation for both the augmented reality markers and the autocomplete suggestions and found them useful for accomplishing the tasks. However, several participants stated that they still wanted a list of names or at least a count of devices. We observed many times during self-directed discovery that users would slowly pan around, moving back and forth to count the teardrop markers and map them to devices. Participant 4 expressed, "Something I want to do now is make sure I really have seen all of the devices in the room. I wish there was a number or something that could tell me. So the best I can do is look at these teardrops." One useful feature might be to change the teardrop color when the device has been shown near the center of the screen, as a way of marking "visited" devices similarly to the way we mark visited hyperlinks.

6.6.3 Knowledge learned from ARticulate enables successful voice interactions, though sensors remain a challenge

After using the ARticulate app, we asked participants to set the phone aside and perform tasks using the voice interface to Alexa. After using the app and switching to voice, people knew what smart (and not smart) actuators were in the room and what their names were, including a light that they had never interacted with in previous tasks and only discovered incidentally. We can see from the large number of ideal scores for the post-app voice phase in Figure 6.7 that information was obtained for Tasks 1-3. We also asked users a new actuator task, Task 5, where users had to figure out to turn on Light A, even though we had not asked users to interact with Light A previously. We can see that though users guessed *Lamp* A about half the time, in keeping with the Floor Lamp and Mood Lamp naming pattern, they were almost all able to accomplish the new task using voice.

However, users still had difficulty with the sensor task. Even after successfully using ARticulate to communicate with Alexa about the sensor, users struggled to remember how to obtain the sensor reading in the final voice phase. They knew it was possible ("I know it's capable because I did it through the app"), and they knew it was a specific phrase ("There was a very specific phrase that was used before"), but participants struggled to remember the details.

Two participants remembered both temperature and Multipurpose Sensor and were able to achieve the goal. Two other participants, Participant 6 and Participant 4, also remembered both pieces of information, but were not able to achieve the goal. Participant 4 said "Tell me the temperature according to Multipurpose Sensor" instead of "What is the temperature according to Multipurpose Sensor," and was frustrated when it did not work. Participant 6 said "What is the temperature...in Multipurpose Sensor?" but Alexa interrupted during the pause and gave her the weather forecast. However, the participant thought Alexa had heard the entire command, so she gave up and did not try again. Of the remaining participants, three obtained the reading by saying something about the temperature "inside" while the remaining two were not able to recall or guess a valid phrase.

Participants appeared particularly resistant to the idea that they needed a proper name for the sensor in order to get a temperature reading. The idea that only the attribute (temperature) should be sufficient was so persistent that even when using ARticulate, Participant 7 manually typed "what is the temperature" when the suggestion showed "what is the temperature according to Multipurpose Sensor." Participant 1 did not think that even temperature should have to be specified, since there was only the one sensor and the one attribute it measured: "For the sensor, if it's the only sensor that is available in the room, maybe just say 'A[lexa], give me a sensor reading' versus give me a *temperature* sensor reading." This assumption that the name Multipurpose Sensor was so unimportant that it was not worth remembering had one notable exception: Participant 8 said before her successful voice attempt, "I'm going to ask her to tell me about the Multipurpose Sensor, because I'm not sure if I just say sensor or temperature she'll know what I'm talking about."

6.6.4 The autocomplete design clashed with the assistant framing

Throughout the sessions, participants consistently anthropomorphized Alexa, and seemed to believe that the devices were mediated through her. For example, instead of wondering aloud "What is that device called?" Participant 2 said, "What is A[lexa] going to think that is called?" This suggests that we successfully framed these smart space tasks as communication tasks with an intelligent assistant, rather than as unmediated device tasks.

However, when it came to the ARticulate app specifically, a few participants suggested that the interaction with Alexa felt like an unnecessary layer of indirection. Participant 4 put the reason into words:

> "I didn't know how to relate to Alexa as a human. [...] I tend to say please and thank you to Siri because she seems like a person. This app was so not—because I couldn't hear her voice anymore—it was a little bit like, are you a robot, are you a person? [...] If I'm supposed to take pictures of things, and it'll give me a list of commands—which was like the core of the app for me, not the texting so much as the list of commands, because I didn't have the list beforehand—having the whole idea of there even being an Alexa seemed silly to me. If it's just specific commands, why can't I just press a button?"

In other words, our autocomplete suggestions were being perceived more like a menu than a true autocomplete to aid in a conversation with Alexa. You can see in Figure 6.1 and Figure 6.2 that our autocomplete design covered the screen with rigidly repeated template text, and also did not provide suggestions for any paraphrases of these "canonical" forms, even though Alexa would understand them. Even just starting a phrase with "please" like one might do conversationally would prevent suggestions from showing. In this way, the suggestions behaved much more like a menu than an autocomplete widget helping a natural conversation. Participants even tended to scroll and select, rather than type.

However, we described to Participant 4 an alternative autocomplete design that waits until you start typing, only shows next-word suggestions as you type rather than displaying everything up front, and provides suggestions for all possible paraphrases. We asked whether this more flexible, more conversational autocomplete would help, and she responded:

> "No, I loved that there was a menu of everything. It was more that the menu made me feel like, why is there this construct called Alexa? [...] I felt like I was communing directly with the smart lamps and stuff. Then it felt weird to have to text Alexa, because I was like, the lamp's right here, Alexa's over there...I'm closer to the lamp than you are! So it felt weird to have to text her about it, when the lamp's right there. [...] It was hard to imagine, why am I asking Alexa to do this, is she like, the lamp whisperer?"

We discuss possible directions for addressing this in Section 6.7.

6.7 Implications

Based on our findings, we now explore three design implications. The first is that we should use discovery-centric designs like ARticulate to help users operate unfamiliar smart spaces. The second is that we need to further explore the tensions between centering intelligent assistants versus centering menus in our designs for discovery. The third is that intelligent assistant designers should recognize that sensors and actuators represent two separate classes of interaction, and natural language interactions with sensors should focus on the data, rather than requiring a device name.

6.7.1 We should use AR and autocomplete-enabled assistant messaging for discovery in unfamiliar smart spaces

Using ARticulate, for the first time users can have successful one-shot natural language interactions with intelligent assistants in an unfamiliar smart spaces. This overcomes the significant challenges posed by the proper device name paradigm, which we characterized in Chapter 3 and Chapter 4. Additionally, users can successfully operate the room using voice afterwards, even to use capabilities that they had not previously interacted with using the messaging app. This means that not only does ARticulate help users accomplish their goals in unfamiliar smart spaces, it also unlocks other "proper device name" interfaces and gives users the option of selecting whichever interface best meets their needs for future tasks. So long as smart space interactions continue to be designed around a proper device name paradigm, the ecosystem of smart space interfaces will benefit from having a discovery-oriented design like ARticulate. Now that users can quickly discover and interact with technologies in an unfamiliar smart space, these technologies can scale and become truly ubiquitous. Further, these physical context-aware messaging options should be integrated natively into the assistant

apps on smartphones. The ability to photograph items annotated with augmented information and chat with an assistant about them could be an interaction style with potential benefits that extend far beyond smart spaces.

6.7.2 We should explore designs around assistants and menus

Given that systems are currently programmed around a device-oriented paradigm, but could potentially benefit from a stronger assistant framing, it is hard to know whether we should take the ARticulate design in the direction of a more realistic conversational autocomplete, or in the direction of a better menu.

A more conversational autocomplete that helps make the language interactions feel more natural would show fewer words at a time, and therefore we would need an alternative way to provide users with an overview of the possible set of device capabilities (e.g., color, brightness, etc.) One option might be that in addition to the device name, devices could display augmented reality "capability icons" that indicate that they can do color and brightness and so on. Then the user knows when they snap the picture not only that the device is smart, but broadly what it is capable of, and can use the more minimal (but also more powerful) autocomplete to simply get the proper names of states (like "lavender blush") as they type.

An improved autocomplete design that leans more in the direction of a tappable menu would follow the advice of several participants and start suggestions for a new photo with a drill-down list of all the device names in the picture. Users would then select a device name to reveal another drill-down list, this time of the device's capabilities, that users would again expand to show the individual canonical utterances. To help build language skills and make assistant interactions feel more natural, upon selecting an utterance, the suggestions could display a list of equivalent paraphrases that the assistant would also understand. While this device-oriented menu direction would no doubt be effective in the short term, it would not be future-proof if assistants trend more towards supporting device-agnostic high-level goals. More work is needed to explore the trade-offs between these different directions.

6.7.3 We should revisit the natural language approach to sensors

The experiments with ARticulate revealed a potentially deeper issue with the underlying natural language understanding system itself. It would appear from our results that sensors require a different natural language approach than actuators. Hearkening back to the interaction framing discussion in Chapter 3, sensor interactions may more intuitively follow a data-oriented, rather than a device-oriented, conceptual model. Ideally, users would not need to specify a specific device at all to receive useful information. While for large deployments this may require sophisticated aggregation techniques or understanding of deployment context, for simple deployments with only a single device that measures a particular attribute, it should be trivial for Alexa to infer which device should be queried to answer the user—the only one available.

We also know from Chapter 3 that moving towards an assistant-mediated data approach will also change the kinds of interactions that people will expect to have with the assistant. Users will often not be interested directly in what the data from a particular device is, but rather in higher-level concepts like "when someone enters the room," or "when the room is empty," or "how much water have I used this month?" Such a system will be much more valuable to users, but the assistant will need to be redesigned to be able to handle these kinds of interactions.

While the above implications apply generally to all intelligent assistants that currently anchor their smart space interactions around proper device names, we also identified an issue perhaps more specific to Alexa. The specificity of the sensor commands that Alexa understands broke the illusion of human-like communication with an intelligent assistant. After Participant 4 discovered that the phrase "Tell me the temperature according to Multipurpose Sensor" does not work, even though "What is the temperature according to Multipurpose Sensor?" does, she expressed: "I feel very frustrated, because when Alexa talks like a person, I feel like I should have a wide net of possible things I can say about asking for the temperature, but because what I have to say is so narrow, it makes me feel like I am becoming more robot-like in order to talk to her, and I think that is not okay." These issues with Alexa demonstrate how natural language interfaces can be fragile in a way that negatively impacts the user experience.

6.8 Summary

In this chapter we were able to finally achieve our goal of one-shot interactions in unfamiliar smart spaces with an agent-mediated interaction framing. We took our previous approach of using autocomplete-aided chat with an intelligent assistant, and added photo messaging, which allowed us to display augmented reality labels for smart devices as a key discovery mechanism to address the precision problem. To evaluate this design, we developed a smartphone app called ARticulate and ran nine user evaluations in a smart home testbed. Presented with an unfamiliar smart space, ARticulate users were able to discover available functionality, including what was *not* available, and accomplish goals with a single interaction attempt. The knowledge users gained from use of ARticulate also translated into the ability to use the voice interface afterwards without a discovery aid. ARticulate is the first interface that makes unfamiliar smart spaces usable by overcoming the challenges posed by the proper device name paradigm. With interfaces like ARticulate, anyone who knows how to message can walk into any smart space and start immediately using it with confidence.

Chapter 7

Conclusion and Future Work

This dissertation has deeply examined the issue of discovery in unfamiliar smart spaces designed around the proper device name paradigm and proposed a solution for supporting successful one-shot interactions with intelligent assistants in such spaces. We found that we could design tools that enabled one-shot interactions, enabling a much more usable future world where assistants could be in any space and part of casual life. However, before this future can be fully realized, there are infrastructure and deployment challenges that must be addressed.

Further, past studies have shown that despite the prevalence of the proper device name paradigm, proper names are not how occupants instinctively refer to smart objects when talking to smart home assistants freed from technical constraints [87]. Over the long term, it is worth exploring more intuitive ways of interacting with intelligent smart spaces that do not require invoking device names and more closely resemble how humans would communicate with each other.

Finally, there are natural extensions of our work in the direction of designing assistants that support other key smart home tasks, such as creating automation rules and debugging, and in enabling assistants to learn and grow their vocabulary through interactions with occupants. We discuss these avenues for future research below.

7.1 Deployment Challenges for Ubiquitous Assistants

This dissertation focused on making smart space assistants usable in unfamiliar smart spaces. But that mostly makes sense in a future where smart spaces and assistants are ubiquitous and can be found in many of the places we go. To deploy these solutions at scale would require infrastructure for sharing augmented reality scans and localization, as well as methods for programmatically exposing what assistants understand for the purposes of autocomplete.

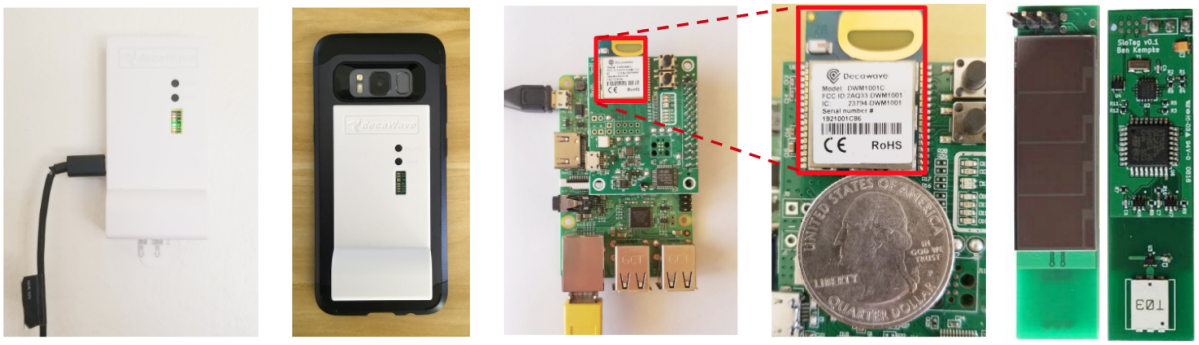


Figure 7.1: High-resolution ultra-wideband (UWB) localization systems. The four pictures on the left show the Decawave MDEK1001 anchor-and-tag system and gateway that provides 10 Hz location updates for mobile devices that move quickly or frequently, such as mobile phones. Modern smartphones have UWB radios built in. On the right is a Slocalization tag, an ultra-low energy UWB tag that does not require batteries.

7.1.1 Augmented reality scans and localization

We envision a future where during initial device setup, the system maintainer would somehow add the device to a digital map of the room. This map would contain information about the physical locations of all the smart devices. When new users entered the unfamiliar smart space, they need only open the app, and the phone should both 1) receive the map information, and 2) be able to localize itself in the room in order to appropriately align the virtual map onto the physical devices—this is called "registration." While sharing and receiving the map information is solvable with well-designed software services, registration for augmented reality is currently an open problem.

The ARticulate study experience revealed the limitations of using a vision-based method for registration (see Figure 6.3). Scans could take anywhere from 15 minutes to hours depending on the lighting conditions and the visual complexity of surfaces in the room, which could cause the phone to lose track of its location in the environment. Twice, users had to wait for us to redo the scan completely or deal with particularly fragile localization due to changing light conditions. This was a problem when performing scans during sunset, where every fifteen minutes the lighting conditions changed enough to completely prevent registration. Turning lights on and off during the session also occasionally changed the lighting conditions enough for the system to lose tracking. We ended up addressing these problems by running systems during daylight hours or at night, and by doing scans in rooms with permanently good lighting that would not be significantly affected by the testbed devices.

Despite the deployment difficulties that augmented reality posed during the study, we still believe that this form of augmented reality feedback will be a feasible solution in the near future. Flagship smartphone manufacturers like Apple and Samsung have released smartphone models with ultra-wideband (UWB) radios that support high-resolution device

localization and orientation, and lidar for creating depth scans of the environment. UWB has been used for fast, high-resolution tracking of mobile devices, and low-power backscatter UWB tags can be used to provide localization for objects that move infrequently [152, 153]. This additional instrumentation has the potential to significantly increase the robustness of location tracking and registration for mobile augmented reality going forward.

7.1.2 APIs for exposing assistant understanding

For an interface like ARticulate to work ubiquitously, it also need a method for receiving autocomplete suggestions that reflect what the smart space assistant for that room understands. Ideally, this would require participation from the major companies that develop intelligent assistants. In the ARticulate study, we built an entire text-based clone of Alexa for smart space interactions, since Alexa does not provide a means to submit text utterances and receive text responses. However, the clone did not always reflect the voice-based Alexa's behavior. It would be better for the consistency of the user experience if the assistants that users message were the same as the voice-based smart space assistants.

This places responsibility on intelligent assistant companies to be able to represent what their assistant understands, which can be difficult given the popularity of black-box models like neural networks. However, one example of an architecture that supports this can be found in the paper "Building a Semantic Parser Overnight" [137]. The authors outline a design methodology for quickly building semantic parsers for new domains. For each capability in the domain, such as turning on a particular device, the developer uses templates to construct a "canonical utterance" in natural language that corresponds to it. To then train the system to understand paraphrases, the developer sends each of the canonical phrases out to a crowdsourcing platform to get a large number of paraphrases for each canonical form. Those paraphrases are then used to train a machine learning system to match a paraphrase to a canonical form, and therefore to an action in the domain. The result is a powerful natural language parser that also has a spanning set of canonical natural language utterances that is guaranteed to cover its functionality. To speed up the development process of these systems, the authors released a toolkit called SEMPRE.

Almond is an IoT virtual assistant built on the SEMPRE architecture [154]. The authors used crowdsourcing to train a parser to understand various smart home commands and queries. The system translated natural language input into a domain-specific language called ThingTalk that could be executed by a controller that interfaced with the available devices and services. While the parser struggled to understand open-ended sentences, when users were aided by first seeing a "cheat sheet" of what the assistant could understand, Almond successfully parsed 80% of the simple phrases and 41% of the compound phrases that users provided.

Our ARticulate implementation also supports the feasibility of this general approach in smart spaces. We wrote a template grammar and populated the possible names and states by discovering the device capabilities through the SmartThings API. This required manually constructing a small type system that allowed us to convert the device capability descriptions to

natural language, but ultimately this allowed us to demonstrate that autocomplete suggestions could be dynamically assembled from system APIs. Though the templates provided users with a canonical set of options, the assistant we built could understand arbitrary paraphrases.

While less than ideal due to inconsistencies in the user experience and the cost of managing separate infrastructure, our implementation demonstrates that it is possible for a third party to construct these "parallel shadow assistants" in order to provide text interactions and autocomplete services if the companies do not incorporate it natively into the intelligent assistants.

7.2 Beyond the Proper Name Paradigm

The proper device name paradigm is a leaky abstraction that has come up from the system design and become a part of smart phone control apps and then intelligent assistants, even though it is not how people intuitively refer to devices. We need to move towards multi-modal, contextually aware interactions with smart space assistants that are able to combine words with other information like gaze direction, pointing, or contextual information to resolve pronouns. We also need to move towards supporting interoperability and applications that are more centered around high-level goals and behaviors and less tied to individual devices. Finally, we need to support more assistant initiative for asking the humans to demonstrate commands, so that the assistant can learn new names and phrases. Assistants start with a baseline mapping of language to action, but if we provide a process for the assistant to extend that language and acquire even more language understanding, assistants could become powerful and personalized to each space.

7.2.1 Multi-modal entity resolution

The first step in re-evaluating how we approach smart home assistant interactions is to combine verbal utterances with contextual information in order to resolve entity references, instead of relying on proper names. In our study of ARticulate, we had participants who expressed that they should be able to tell Alexa to turn off the only light that is on, and that Alexa should be able to figure out which device they're referring to if they say "sensor" and there is only one sensor in the room. Similarly, saying "Turn off the light" when only one light would be completely natural when speaking to another human, but is not currently supported. Existing smart home assistants, such as Alexa, have the ability to resolve these types of references simply by querying the states and types of the devices that they already have access to through their cloud endpoints.

An additional source of information that is not currently captured but which would be invaluable is gaze direction during the utterance. Users say generic phrases like "Turn off the light," but in the vast majority of cases they look at the light they are referring to [87]. In a way, ARticulate works by capturing gaze information. However, there could be many other

approaches to gaze detection. Gaze combined with the utterance should be enough to resolve entity references in the vast majority of cases, without use of any proper names.

In Wizard-of-Oz studies where secret human researchers are able to simulate more capable smart home assistants, participants also use location or relational phrases such "turn off the light in the corner" or "turn on the light to the left of the TV" [87]. This contextual information could potential be solved with more sensing or data collection about object locations. However, assistants could also learn which device these phrases refer to—without any additional sensing—by asking the user to demonstrate the correct action, which we discuss next.

7.2.2 Assistants with learning initiative

Smart space assistants currently act as passive translation layers for human intention. As illustrated in Figure 7.2, human users hold a mental model of the world, and to place the world into a desired goal state, they communicate to the assistant, which directly converts their natural language utterance into an action on the world. The human observes the result, and repeats as needed. In this workflow, the assistant is little more than the natural language equivalent of a button interface.

However, if the assistant is also given a mental model of the world and the agency to act, then many more powerful workflows emerge that allow the assistant to learn new capabilities, entities, and phrases. The assistant can act on the *human*, such as by asking them to perform a task, who then interacts with the world, which the assistant can observe. This means that when a user uses a command or device reference that the smart space assistant does not understand, the assistant could ask the human to demonstrate it using one of the other smart space interfaces, and thus acquire the understanding of a new phrase. Concretely, the assistant could learn which device the phrase the "light in the corner" refers to, or what "when I leave the house" looks like from a sensor standpoint. Similarly, the human can interact with the world directly, such as by using another smart space app, and the assistant can observe both the user's actions and the events leading up to it. This would enable imitation learning, where the assistant learns to anticipate and perform smart space actions that the user would normally perform. The assistant can take initiative and interact with the world on its own, or the assistant and human can interact with each other. This way of thinking echoes themes in mixed initiative interfaces [155–158], agent interfaces [82, 159–161], and task-oriented reinforcement learning [34], especially when tasks are specified by natural language [162–167]. One ubiquitous computing system, Roadie, provides an example implementation of such an approach, by mapping users goals to device functionality and using a partial planner and mixed-initiative AI to help users debug [168]. Moving in this direction would allow for a future where smart home intelligent assistants become more extensible, powerful, and personalized.

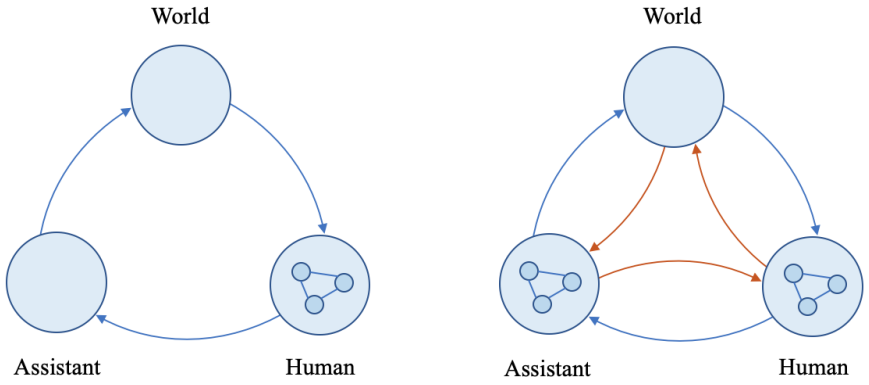


Figure 7.2: The present and future of intelligent agents in smart spaces. In the current model, a human user maintains a mental model of the world. To achieve their desired goal state in the world, they initiate an interaction with the agent, which exists merely as a pass-through for translating the human's interaction into a command that affects the world. The human observes the new state of the world and repeats until the goal state is reached. However, if we consider the agent as also having a mental model of the world and a similar level of agency and initiative, we are put into the paradigm on the right. In this paradigm there are many kinds of possible interactions between agent and human. Many of these workflows are critical for agents to learn new phrases, new functionality, and user preferences.

7.2.3 Language-free assistant interfaces

Another option for avoiding the problem of proper names is by not using language at all to interact with intelligent assistants. Reliance on language comes with downsides to accessibility. A large percent of households in the US do not speak English at home. At least one participant in the ARticulate study, Participant 6, had a strong accent and reported that she stopped using intelligent assistants because they had trouble understanding her. Additionally, children are a significant category of smart home passenger users, but they often struggle with language interfaces. There is a window of development where children may want to operate basic functionality of their home, but may lack reading and writing proficiency, and additionally face difficulties being understood verbally by intelligent assistants. While no doubt speech recognition technology will continue to progress, we feel there is fertile unexplored territory in language-free smart home assistants.

The key metaphor to draw on for language-free smart home assistant design is that of *animism*. All around the world early human societies tapped in to animistic explanations for why their natural environment behaved the way that it did, a proclivity that has continued through the modern day – contemporary studies have found that humans frequently ascribe animistic characteristics to technology [169–171]. While early anthropological work framed animism as an explanatory framework that is fundamentally in error about the way the world truly operates, recent work has recast animism as a valid way to conceptualize our relationships with the objects around us [172]. Drawing on this new line of thinking, several

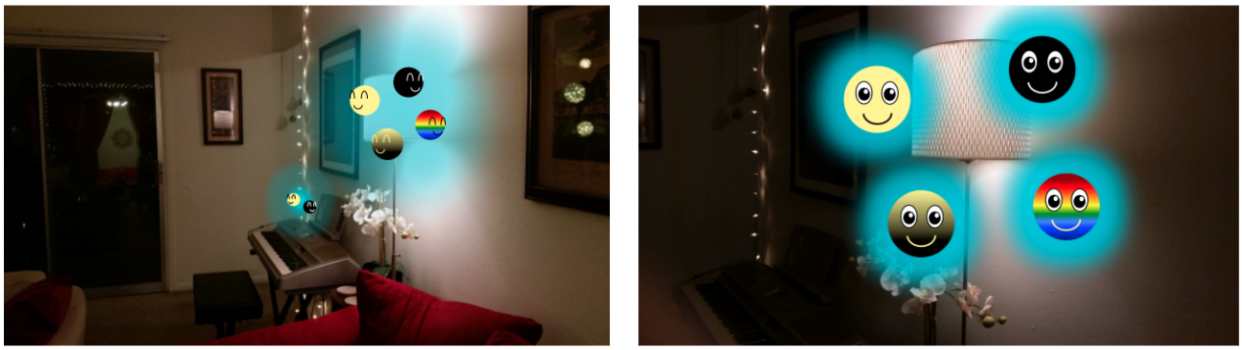


Figure 7.3: Pixie augmented reality interface supporting discovery and invocation of smart space functionality. The left image illustrates the exposed functionality of two smart devices: a smart switch that can be turned on and off, and a smart light that also has dimming and color changing ability. The pixies have their eyes closed so that residents do not feel socially uncomfortable. The right image shows the pixies opening their eyes and engaging with the user as the user approaches and they come into "focus." Functionality is invoked by pinching or pinch-and-dragging the pixies, who giggle like it tickles. Pixies are anthropomorphized, like current smart space assistants, but are language-free, making them more widely accessible to populations like children and residents who prefer to speak other languages at home.

HCI researchers have proposed using animism in the design of smart spaces and connected devices [173–178].

To design language-free smart home assistants, we can draw on animistic archetypes of wild animals and spirits – entities that are intelligent, privately motivated, and may have messages they want to communicate, but which may not speak human language. One example design is what we call pixies. The pixies are a concept for an augmented reality interface that also helps with affordance discovery like ARticulate, but is language-free (and thus more accessible). The idea is that each smart device is represented as a "magical hotspot" that attracts "pixies." Each pixie represents a particular action that can be taken, such as dimming, changing the color, or turning the device on and off. The idea to standardize around actions, or verbs, instead of devices builds off of prior work that found that while smart home users employ many unique nouns in their commands, they use very few unique verbs [87]. This design choice means that even when encountering a novel smart device, the user will be able to understand and use the familiar actions (such as on and off).

Figure 7.3 shows an example of pixies in the ambient environment, and pixies when the user approaches the magical area and they come into focus. To invoke the device functionality, users grasp or pinch the pixies, who respond as though the interaction tickled pleasantly. By pinching the on or off pixie, the device will turn on or off. By pinch-and-dragging the dimming or hue pixie, the user can change qualities of the light. This leads to natural interactions, such as throwing the hue pixie across the room to cause the light to scroll through the rainbow.

Though this illustration shows the potential direction of language-free smart home assis-

tants inspired by concepts from animism, there are more design challenges to address. To bring this interface into functional parity with existing device-oriented app would require determining how to manipulate groups of devices together, how to represent the concept of scenes, and how to represent ambient qualities such as temperature or humidity or whole-home energy usage. Video feeds from networked cameras have also not been addressed, but could perhaps draw upon a "magic mirror" or scrying metaphor.

7.3 Assistants for Automation and Debugging

In this dissertation we only evaluated systems where the assistant performed "immediate actions," which result in an immediate change in the environment or an immediate answer about the present moment. However, if assistants were extended to understand time, so that they could represent concepts of events and causality, then the assistant could help with two significant smart home activities: authoring automation rules, and debugging.

7.3.1 Assistants for automation rules

Authoring automation rules is a significant activity performed by smart home users. However, humans are quite bad at authoring smart home automation rules. Recent work has outlined common cognitive errors that people make while crafting automation rules [179–181]. Assistants could dramatically shorten the iteration cycle and help humans make rules work as expected the first time by looking for these common errors. For example, if a user sets a rule that turns the lights on, an assistant could ask if they would like a rule that also turns the lights off. Alternatively, the assistant could ask the user what they expect the rule to do in a set of cleverly selected scenarios, and then offer suggestions for how to change the rule to do what they want in the various scenarios.

7.3.2 Assistants for debugging

Debugging is strongly related to automation rules. As soon the system starts behaving autonomously, users will want to know why things are happening. It would be helpful during debugging to be able to credit a behavior to the rule that caused it. Approaches to helping users understand the behavior of their system can draw inspiration from past work on explanation-oriented interfaces like Whyline [182], the principles of which have been applied to some ubiquitous computing contexts by Lin and Dey [183]. Fruitful areas for exploration include making sure that explanations occur at the appropriate level of detail, combining the partial knowledge that the assistant has access to with the partial knowledge that the human has access to, and collaboratively guiding users through hypothesis testing to begin systematically ruling out potential explanations.

7.4 When the Unfamiliar Becomes Familiar

The goal of this dissertation was to understand and address a major challenge facing smart spaces as they become more ubiquitous and users have more frequent encounters with unfamiliar smart spaces that they did not set up themselves. We observed that the proper device name paradigm that smart spaces are currently designed around poses a significant obstacle to usability when users first enter an unfamiliar smart space.

To systematically explore the effects of the proper name paradigm on user discovery, we ran two studies. The first examined the effects of device-centrism, and found that framing interactions around a collection of smart devices resulted in limited user expectations of automation and interoperability. However, adding an intelligent assistant as a mediating layer on top of the devices was able to mitigate some of the limiting effects on end user mental models. This means that to encourage users to perceive smart spaces as supporting valuable high-level applications, device-centric smart space interfaces should at least use an intelligent assistant interaction framing. The second study examined proper names, and found that not only were proper names for smart devices inherently difficult to guess, even when given a proper name, users may struggle to determine which real-world device the name refers to. This means that providing users with a list of device names will only bound their trial-and-error guesses—it will not enable one-shot interactions or help users understand when a goal is *not* possible.

Based on the results of those two studies, we designed a messaging interface for communicating with intelligent smart space assistants that enables users in unfamiliar smart spaces to accomplish their goals on the first attempt. Further, after a few uses, this interface unlocks the ability to use voice and potentially other name-centric smart space interfaces. The key discovery mechanisms that we used are augmented reality markers for smart devices, and autocomplete suggestions for composing the message to the assistant. This design bring us closer to a future in which anyone can walk into an unfamiliar smart space and immediately interact with it as if messaging an old friend.

Bibliography

[1] Frances K Aldrich. "Smart homes: Past, present and future". In: *Inside the Smart Home*. Springer, 2003, pp. 17–39.

[2] Lynn Spigel. "Smart homes: Digital lifestyles practiced and imagined". In: *Relocating Television*. Routledge, 2012, pp. 260–278.

[3] Lynn Spigel. "Media homes: Then and now". In: *International Journal of Cultural Studies* 4.4 (2001), pp. 385–411.

[4] Edward Cline and Buster Keaton. *The Electric House*. First National Pictures, 1922.

[5] Phyllis Palmer. *Domesticity and dirt: Housewives and domestic servants in the United States, 1920-1945*. Temple University Press, 2010.

[6] Dag Spicer. *The Echo IV home computer: 50 years later*. 2016. URL: https://computerhistory.org/blog/the-echo-iv-home-computer-50-years-later (visited on 01/29/2021).

[7] Women's Bureau. *Labor force participation rate of mothers by age of youngest child, March 1975-2016*. U.S. Department of Labor. URL: https://www.dol.gov/agencies/wb/data/facts-over-time/women-in-the-labor-force#labor-force-participation-rate-of-mothers-by-age-of-youngest-child.

[8] Bin Zhou, Wentao Li, Ka Wing Chan, Yijia Cao, Yonghong Kuang, Xi Liu, and Xiong Wang. "Smart home energy management systems: Concept, configurations, and scheduling strategies". In: *Renewable and Sustainable Energy Reviews* 61 (2016), pp. 30–40.

[9] AJ Bernheim Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. "Home automation in the wild: challenges and opportunities". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2011, pp. 2115–2124.

[10] Rosslin John Robles and Tai-hoon Kim. "Applications, systems and methods in smart home technology: a review". In: *International Journal of Advanced Science And Technology* 15 (2010), pp. 37–48.

[11] Michael Miller. *The Internet of Things: How smart TVs, smart cars, smart homes, and smart cities are changing the world*. Pearson Education, 2015.

[12] Rosslin John Robles and Tai-hoon Kim. "A review on security in smart home development". In: *International Journal of Advanced Science and Technology* 15 (2010).

[13] Richard Harper. "Inside the smart home: Ideas, possibilities and methods". In: *Inside the Smart Home*. Springer, 2003, pp. 1–13.

[14] Jie Wan, Xiang Gu, Liang Chen, and Jin Wang. "Internet of Things for ambient assisted living: Challenges and future opportunities". In: *2017 International Conference on Cyber-enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE. 2017, pp. 354–357.

[15] Debajyoti Pal, Tuul Triyason, and Suree Funikul. "Smart homes and quality of life for the elderly: a systematic review". In: *2017 IEEE International Symposium on Multimedia (ISM)*. IEEE. 2017, pp. 413–419.

[16] Michael Todd and Sreela Sasi. "Intelligent virtual companion system for independent living". In: *IC-AI*. Citeseer. 2006, pp. 439–445.

[17] Thomas Zachariah, Noah Klugman, Bradford Campbell, Joshua Adkins, Neal Jackson, and Prabal Dutta. "The Internet of Things has a gateway problem". In: *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*. 2015, pp. 27–32.

[18] Bin Yuan, Yan Jia, Luyi Xing, Dongfang Zhao, XiaoFeng Wang, and Yuqing Zhang. "Shattered chain of trust: Understanding security risks in cross-cloud iot access delegation". In: *29th USENIX Security Symposium*. 2020, pp. 1183–1200.

[19] Pew Research Center. *Nearly half of Americans use digital voice assistants, mostly on their smartphones*. Washington, D.C., 2017. URL: https://www.pewresearch.org/fact-tank/2017/12/12/nearly-half-of-americans-use-digital-voice-assistants-mostly-on-their-smartphones.

[20] Tawfiq Ammari, Jofish Kaye, Janice Y Tsai, and Frank Bentley. "Music, search, and IoT: How people (really) use voice assistants." In: *ACM Transactions on Computer-Human Interaction* 26.3 (2019), pp. 17–1.

[21] Amy He. *Amazon maintains convincing lead in US smart speaker market*. 2020. URL: https://www.emarketer.com/content/amazon-maintains-convincing-lead-in-us-smart-speaker-market (visited on 07/06/2021).

[22] Vinay Koshy, Joon Sung Sung Park, Ti-Chung Cheng, and Karrie Karahalios. ""We just use what they give us": Understanding passenger user perspectives in smart homes". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–14.

[23] Christine Geeng and Franziska Roesner. "Who's in control? Interactions in multi-user smart homes". In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–13.

[24] Sarah Mennicken and Elaine M Huang. "Hacking the natural habitat: An in-the-wild study of smart homes, their development, and the people who live in them". In: *International Conference on Pervasive Computing*. Springer. 2012, pp. 143–160.

[25] Josephine Lau, Benjamin Zimmerman, and Florian Schaub. "Alexa, are you listening? Privacy perceptions, concerns and privacy-seeking behaviors with smart speakers". In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW (2018), pp. 1–31.

[26] Sophie Nyborg. "Pilot users and their families: Inventing flexible practices in the smart grid". In: *Science & Technology Studies* (2015).

[27] Jong-bum Woo and Youn-kyung Lim. "User experience in do-it-yourself-style smart homes". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015, pp. 779–790.

[28] Ryen W White. "Skill discovery in virtual assistants". In: *Communications of the ACM* 61.11 (2018), pp. 106–113.

[29] Computer History Museum. *Bytes for Bites: The Kitchen Computer*. URL: https://www.computerhistory.org/revolution/minicomputers/11/362 (visited on 01/29/2021).

[30] Daniela Hernandez. *Before the iPad, there was the Honeywell Kitchen Computer*. 2012. URL: https://www.wired.com/2012/11/kitchen-computer (visited on 01/29/2021).

[31] Bill Gates, Nathan Myhrvold, Peter Rinearson, and Donald Domonkos. *The Road Ahead*. Viking New York, 1995.

[32] Mark Weiser. "The computer for the 21st century". In: *ACM SIGMOBILE Mobile Computing and Communications Review* 3.3 (1999), pp. 3–11.

[33] Debashis Saha and Amitava Mukherjee. "Pervasive computing: a paradigm for the 21st century". In: *Computer* 36.3 (2003), pp. 25–31.

[34] Michael C Mozer. "The Neural Network House: An environment hat adapts to its inhabitants". In: *Proceedings of the AAAI Spring Symposium on Intelligent Environments*. 1998, pp. 110–114.

[35] Cory D Kidd, Robert Orr, Gregory D Abowd, Christopher G Atkeson, Irfan A Essa, Blair MacIntyre, Elizabeth Mynatt, Thad E Starner, and Wendy Newstetter. "The Aware Home: A living laboratory for ubiquitous computing research". In: *International Workshop on Cooperative Buildings*. Springer. 1999, pp. 191–198.

[36] Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. "The Gator Tech Smart House: A programmable pervasive space". In: *Computer* 38.3 (2005), pp. 50–60.

[37] Michael H Coen et al. "Design principles for intelligent environments". In: *AAAI/IAAI*. 1998, pp. 547–554.

[38] Stephen S Intille. "Designing a home of the future". In: *IEEE Pervasive Computing* 1.2 (2002), pp. 76–82.

[39] Brad Johanson, Armando Fox, and Terry Winograd. "The interactive workspaces project: Experiences with ubiquitous computing rooms". In: *IEEE Pervasive Computing* 1.2 (2002), pp. 67–74.

[40] W Keith Edwards and Rebecca E Grinter. "At home with ubiquitous computing: seven challenges". In: *LNCS* 2201 (2001), pp. 256–272.

[41] Gregory D Abowd and Elizabeth D Mynatt. "Charting past, present, and future research in ubiquitous computing". In: *ACM Transactions on Computer-Human Interaction (TOCHI)* 7.1 (2000), pp. 29–58.

[42] Sumi Helal. "Programming pervasive spaces". In: *Pervasive Computing, IEEE* 4.1 (2005), pp. 84–87.

[43] Hiroshi Ishii and Brygg Ullmer. "Tangible Bits: Towards seamless interfaces between people, bits and atoms". In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 1997, pp. 234–241.

[44] Khai N Truong, Elaine M Huang, and Gregory D Abowd. "CAMP: A magnetic poetry interface for end-user programming of capture applications for the home". In: *International Conference on Ubiquitous Computing*. 2004.

[45] Anind K Dey, Timothy Sohn, Sara Streng, and Justin Kodama. "iCAP: Interactive prototyping of context-aware applications". In: *Pervasive Computing*. Springer, 2006, pp. 254–271.

[46] Jan Humble, Andy Crabtree, Terry Hemmings, Karl-Petter Åkesson, Boriana Koleva, Tom Rodden, and Pär Hansson. "'Playing with the bits': User-configuration of ubiquitous domestic environments". In: *International Conference on Ubiquitous Computing*.

[47] Tom Rodden, Andy Crabtree, Terry Hemmings, Boriana Koleva, Jan Humble, Karl-Petter Åkesson, and Pär Hansson. "Between the dazzle of a new building and its eventual corpse: Assembling the ubiquitous home". In: *Proceedings of 2004 ACM Syposium on Designing Interactive Systems*. 2004.

[48] W Keith Edwards, Mark W Newman, Jana Z Sedivy, Trevor F Smith, Dirk Balfanz, Diana K Smetters, H Chi Wong, and Shahram Izadi. "Using Speakeasy for ad hoc peer-to-peer collaboration". In: *Proceedings of the 2002 ACM Conference on Computer-Supported Cooperative Work*. 2002, pp. 256–265.

[49] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: a survey". In: *Computer Networks* 54.15 (2010), pp. 2787–2805.

[50] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. "The Internet of Things—a survey of topics and trends". In: *Information Systems Frontiers* 17.2 (2015), pp. 261–274.

[51] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. "Internet of Things: A survey on enabling technologies, protocols, and applications". In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2347–2376.

[52] Farzad Samie, Lars Bauer, and Jörg Henkel. "IoT technologies for embedded computing: a survey". In: *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*. IEEE. 2016, pp. 1–10.

[53] Mark T Bohr and Ian A Young. "CMOS scaling trends and beyond". In: *IEEE Micro* 37.6 (2017), pp. 20–29.

[54] Yifan Sun, Nicolas Bohm Agostini, Shi Dong, and David Kaeli. "Summarizing CPU and GPU design trends with product data". In: *arXiv preprint arXiv:1911.11313* (2019).

[55] Neal Jackson, Joshua Adkins, and Prabal Dutta. "Capacity over capacitance for reliable energy harvesting sensors". In: *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. 2019, pp. 193–204.

[56] Flora Salim and Usman Haque. "Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and Internet of Things". In: *International Journal of Human-Computer Studies* 81 (2015), pp. 31–48.

[57] Clement Delangue. *How the revolution of natural language processing is changing the way companies understand text*. 2020. URL: https://techcrunch.com/sponsor/nvidia/how-the-revolution-of-natural-language-processing-is-changing-the-way-companies-understand-text.

[58] Mehdi Assefi, Mike Wittie, and Allan Knight. "Impact of network performance on cloud speech recognition". In: *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE. 2015, pp. 1–6.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[60] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[61] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *arXiv preprint arXiv:2005.14165* (2020).

[62] Don Norman. *The Design of Everyday Things: Revised and expanded edition*. Constellation, 2013.

[63] Meghan Clark, Mark W. Newman, and Prabal Dutta. "Devices and data and agents, oh my: How smart home abstractions prime end-User mental models". In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (Sept. 2017), 44:1–44:26. ISSN: 2474-9567. DOI: 10.1145/3132031. URL: http://doi.acm.org/10.1145/3132031.

[64] Jerome S Bruner. "The act of discovery". In: *Harvard Educational Review* (1961).

[65] John M Carroll and Mary Beth Rosson. "Paradox of the active user". In: *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*. 1987, pp. 80–111.

[66] Eric Corbett and Astrid Weber. "What can I say? Addressing user experience challenges of a mobile voice user interface for accessibility". In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 2016, pp. 72–82.

[67] Brenda Dervin. "An overview of sense-making research: concepts, methods and results to date". In: *International Communications Association Annual Meeting*. 1983.

[68] Daniel M Russell, Mark J Stefik, Peter Pirolli, and Stuart K Card. "The cost structure of sensemaking". In: *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*. 1993, pp. 269–276.

[69] Peter A. Bibby and Stephen J. Payne. "Instruction and practice in learning to use a device". In: *Cognitive Science* 20.4 (Aug. 1996), pp. 539–578.

[70] M Granger Morgan. *Risk communication: a mental models approach*. Cambridge University Press, 2002.

[71] Ann Bostrom, Cynthia J Atman, Baruch Fischhoff, and M Granger Morgan. "Evaluating risk communications: Completing and correcting mental models of hazardous processes, Part II". In: *Risk Analysis* 14.5 (1994), pp. 789–798.

[72] Jörg Niewöhner, Patrick Cox, Simon Gerrard, and Nick Pidgeon. "Evaluating the efficacy of a mental models approach for improving occupational chemical risk protection". In: *Risk Analysis* 24.2 (2004), pp. 349–361.

[73] Stephen J Payne, Helen R Squibb, and Andrew Howes. "The nature of device models: The yoked state space hypothesis and some experiments with text editors". In: *Human-Computer Interaction* 5.4 (1990), pp. 415–444.

[74] Anind K. Dey, Timothy Sohn, Sara Streng, and Justin Kodama. "iCAP: Interactive prototyping of context-aware applications". In: *International Conference on Pervasive Computing*. 2006. ISBN: 3540338942. DOI: 10.1007/11748625_16.

[75] Blase Ur, Elyse Mcmanus, Melwyn Pak, Yong Ho, and Michael L Littman. "Practical trigger-action programming in the smart home". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014, pp. 803–812. DOI: 10.1145/2556288.2557420.

[76] George W. Furnas, Thomas K. Landauer, Louis M. Gomez, and Susan T. Dumais. "The vocabulary problem in human-system communication". In: *Communications of the ACM* 30.11 (1987), pp. 964–971.

[77] Arnab Nandi and HV Jagadish. "Guided interaction: Rethinking the query-result paradigm". In: *Proceedings of the VLDB Endowment* 4.12 (2011), pp. 1466–1469.

[78] Axel Liljencrantz. *Friendly Interactive Shell*. 2005. URL: https://fishshell.com (visited on 09/20/2019).

[79] Robert C Miller, Victoria H Chou, Michael Bernstein, Greg Little, Max Van Kleek, David Karger, and MC Schraefel. "Inky: A sloppy command line for the web with rich visual feedback". In: *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*. 2008, pp. 131–140.

[80] Ben Shneiderman. "The future of interactive systems and the emergence of direct manipulation". In: *Behaviour & Information Technology* 1.3 (1982), pp. 237–256.

[81] Edwin L Hutchins, James D Hollan, and Donald A Norman. "Direct manipulation interfaces". In: *Human–computer Interaction* 1.4 (1985), pp. 311–338.

[82] Ben Shneiderman and Pattie Maes. "Direct manipulation vs. interface agents". In: *Interactions* 4.6 (1997), pp. 42–61.

[83] Richard A Bolt. ""Put-that-there": Voice and gesture at the graphics interface". In: *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*. 1980, pp. 262–270.

[84] Andrew Wilson and Hubert Pham. "Pointing in intelligent environments with the WorldCursor". In: *Interact*. Citeseer. 2003.

[85] Sven Mayer, Gierad Laput, and Chris Harrison. "Enhancing mobile voice assistants with WorldGaze". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 2020, pp. 1–10.

[86] Mark W Newman, Jana Z Sedivy, Christine M Neuwirth, W Keith Edwards, Jason I Hong, Shahram Izadi, Karen Marcelo, and Trevor F Smith. "Designing for serendipity: Supporting end-user configuration of ubiquitous computing environments". In: *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*. 2002, pp. 147–156.

[87] Barry Brumitt and Jonathan J Cadiz. "'Let there be light': Examining interfaces for homes of the future". In: *Human Computer Interaction–INTERACT'01*. 2001, pp. 375–82.

[88] James H Bradford. "The human factors of speech-based interfaces: a research agenda". In: *ACM SIGCHI Bulletin* 27.2 (1995), pp. 61–67.

[89] Nicole Yankelovich. "How do users know what to say?" In: *Interactions* 3.6 (1996), pp. 32–43.

[90] Nicole Yankelovich, Gina-Anne Levow, and Matt Marx. "Designing SpeechActs: Issues in speech user interfaces". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1995, pp. 369–376.

[91] Laurent Karsenty. "Shifting the design philosophy of spoken natural language dialogue: From invisible to transparent systems". In: *International Journal of Speech Technology* 5.2 (2002), pp. 147–157.

[92] Fang Chen. *Designing human interface in speech technology*. Springer Science & Business Media, 2006.

[93] Marti A Hearst. "UIs for faceted navigation: Recent advances and remaining open problems". In: *HCIR 2008: Proceedings of the Second Workshop on Human-Computer Interaction and Information Retrieval*. 2008.

[94] Fei Cai, Maarten De Rijke, et al. "A survey of query auto completion in information retrieval". In: *Foundations and Trends in Information Retrieval* 10.4 (2016), pp. 273–363.

[95] Ian H Witten, John G Cleary, and John J Darragh. "The Reactive Keyboard: A new technology for text entry". In: *Converging Technologies: Proceedings of the Canadian Information Processing Society Conference*. 1983, pp. 151–156.

[96] John J. Darragh, Ian H. Witten, and Mark L. James. "The reactive keyboard: a predictive typing aid". In: *Computer* 23.11 (1990), pp. 41–49.

[97] Google. *How Google autocomplete works in Search*. 2019. URL: https://www.blog.google/products/search/how-google-autocomplete-works-search (visited on 02/15/2019).

[98] Korinna Grabski and Tobias Scheffer. "Sentence completion". In: *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM. 2004, pp. 433–439.

[99] Ryen W White and Gary Marchionini. "Examining the effectiveness of real-time query expansion". In: *Information Processing & Management* 43.3 (2007), pp. 685–704.

[100] Google. *Google Assistant*. 2019. URL: https://assistant.google.com (visited on 02/14/2019).

[101] Google. *Google Home*. 2019. URL: https://store.google.com/us/product/google_home (visited on 09/19/2019).

[102] Efthimis N Efthimiadis. "Query expansion". In: *Annual review of information science and technology (ARIST)* 31 (1996), pp. 121–87.

[103] Jurgen Koenemann and Nicholas J Belkin. "A case for interaction: A study of interactive information retrieval behavior and effectiveness". In: *Proceeding of the ACM SIGCHI Conference on Human Factors in Computing Systems*. Citeseer. 1996, pp. 205–212.

[104] Arto Vihavainen, Juha Helminen, and Petri Ihantola. "How novices tackle their first lines of code in an ide: Analysis of programming session traces". In: *Proceedings of the 14th Koli Calling International Conference on Computing Education Research*. ACM. 2014.

[105] Gail C Murphy, Mik Kersten, and Leah Findlater. "How are Java software developers using the Elipse IDE?" In: *IEEE Software* 23.4 (2006), pp. 76–83.

[106] Ivan E Sutherland. "A head-mounted three dimensional display". In: *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*. 1968, pp. 757–764.

[107] Eric Rose, David Breen, Klaus H Ahlers, Chris Crampton, Mihran Tuceryan, Ross Whitaker, and Douglas Greer. "Annotating real-world objects using augmented reality". In: *Computer Graphics*. Elsevier, 1995, pp. 357–370.

[108] Jun Rekimoto and Katashi Nagao. "The world through the computer: Computer-augmented interaction with real world environments". In: *Proceedings of the 8th Annual ACM Symposium on User Interface and Software Technology*. 1995, pp. 29–36.

[109] Steven Feiner, Blair MacIntyre, Tobias Höllerer, and Anthony Webster. "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment". In: *Personal Technologies* 1.4 (1997), pp. 208–217.

[110] Jason Wither, Stephen DiVerdi, and Tobias Höllerer. "Annotation in outdoor augmented reality". In: *Computers & Graphics* 33.6 (2009), pp. 679–689.

[111] Ronald T Azuma. "A survey of augmented reality". In: *Presence: Teleoperators & Virtual Environments* 6.4 (1997), pp. 355–385.

[112] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. "Recent advances in augmented reality". In: *IEEE Computer Graphics and Applications* 21.6 (2001), pp. 34–47.

[113] DWF Van Krevelen and Ronald Poelman. "A survey of augmented reality technologies, applications and limitations". In: *International Journal of Virtual Reality* 9.2 (2010), pp. 1–20.

[114] Riya Aggarwal and Abhishek Singhal. "Augmented reality and its effect on our life". In: *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE. 2019, pp. 510–515.

[115] Dimitris Chatzopoulos, Carlos Bermejo, Zhanpeng Huang, and Pan Hui. "Mobile augmented reality survey: from where we are to where we go". In: *IEEE Access* 5 (2017), pp. 6917–6950.

[116] Franziska Roesner, Brian T Gill, and Tadayoshi Kohno. "Sex, lies, or kittens? Investigating the use of Snapchat's self-destructing messages". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2014, pp. 64–76.

[117] Joseph B Bayer, Nicole B Ellison, Sarita Y Schoenebeck, and Emily B Falk. "Sharing the small moments: Ephemeral social interaction on Snapchat". In: *Information, Communication & Society* 19.7 (2016), pp. 956–977.

[118] Jette Kofoed and Malene Charlotte Larsen. "A snap of intimacy: Photo-sharing practices among young people on social media". In: *First Monday* (2016).

[119] Jenni Niemelä-Nyrhinen and Janne Seppänen. "Visual communion: The photographic image as phatic communication". In: *New Media & Society* 22.6 (2020), pp. 1043–1057.

[120] Adrian A de Freitas, Michael Nebeling, Xiang'Anthony' Chen, Junrui Yang, Akshaye Shreenithi Kirupa Karthikeyan Ranithangam, and Anind K Dey. "Snap-to-it: A user-inspired platform for opportunistic device interactions". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 5909–5920.

[121] Mike Black, Katie Dana, Kim Gaskins, Courtney Gaynier, Al Lemieux, and Kyle McKinley. *The Internet of Things: Can it find a foothold with mainstream audiences today?* Tech. rep. The Nielsen Company, Nov. 2014.

[122] Stephen Payne, Helen Squibb, and Andrew Howes. "The nature of device models: The yoked state space hypothesis and some experiments with text editors". In: *Human-Computer Interaction* 5.4 (Dec. 1990), pp. 415–444.

[123] Roy Want, Bill N Schilit, and Scott Jenson. "Enabling the Internet of Things: The IoT vision". In: *Computer* 48.1 (2015), pp. 28–35.

[124] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. "Internet of Things: A survey on enabling technologies, protocols, and applications". In: *IEEE Communications Surveys and Tutorials* 17.4 (2015), pp. 2347–2376. DOI: 10.1109/COMST.2015.2444095.

[125] Gerd Kortuem, Fahim Kawsar, Vasughi Sundramoorthy, and Daniel Fitton. "Smart objects as building blocks". In: *IEEE Internet Computing* 14.1 (2010), pp. 44–51.

[126] Pankesh Patel and Animesh Pathak. "Towards application development for the Internet of Things". In: *ACM/IFIP/USENIX 12th International Middleware Conference*. 2011, pp. 1145–711266. DOI: 10.1145/2093190.2093195⟩.

[127] Ewa Luger and Abigail Sellen. "'Like having a really bad PA': The gulf between user expectation and experience of conversational agents". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016. DOI: 10.1145/2858036.2858288.

[128] Eric Horvitz. "Principles of mixed-initiative user interfaces". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 1999, pp. 159–166. ISBN: 0201485591. DOI: 10.1145/302979.303030.

[129] David V Keyson, Marc PAJ de Hoogh, Adinda Freudenthal, and Arnold POS Vermeeren. "The Intelligent Thermostat: A mixed-initiative user interface". In: *CHI'00 Extended Abstracts on Human Factors in Computing Systems*. 2000, pp. 59–60.

[130] Devika Pisharoty, U Virginia, Rayoung Yang, and Mark W Newman. "ThermoCoach: Reducing home energy consumption with personalized thermostat recommendations". In: *2nd ACM International Conference on Embedded Systems For Energy-Efficient Built Environments*. 2015. DOI: 10.1145/2821650.2821671.

[131] Amazon.com, Inc. *Amazon Mechanical Turk*. 2015. URL: https://www.mturk.com.

[132] U.S. Census Bureau. *Educational Attainment in the United States: 2015*. U.S. Department of Commerce. Mar. 2016.

[133] OpenJS Foundation. *Node-RED*. 2021. URL: https://nodered.org (visited on 07/12/2021).

[134] François Portet, Michel Vacher, Caroline Golanski, Camille Roux, and Brigitte Meillon. "Design and evaluation of a smart home voice interface for the elderly: Acceptability and objection aspects". In: *Personal Ubiquitous Computing* 17.1 (Jan. 2013), pp. 127–144. DOI: 10.1007/s00779-011-0470-5.

[135] Ewa Luger and Abigail Sellen. "'Like having a really bad PA': The gulf between user expectation and experience of conversational agents". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 5286–5297.

[136] Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. "Maximizing the guessability of symbolic input". In: *CHI '05 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '05. Portland, OR, USA: Association for Computing Machinery, 2005, pp. 1869–1872. DOI: 10.1145/1056808.1057043.

[137] Yushi Wang, Jonathan Berant, and Percy Liang. "Building a semantic parser overnight". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2015, pp. 1332–1342.

[138] Terence J. Parr and Russell W. Quong. "ANTLR: A predicated-LL (k) parser generator". In: *Software: Practice and Experience* 25.7 (1995), pp. 789–810.

[139] John Brooke. "SUS: A quick and dirty usability scale". In: *Usability Evaluation in Industry* 189 (1996).

[140] Joanne K Rowling. *Harry Potter and the Prisoner of Azkaban*. Bloomsbury Publishing, 1999.

[141] Eve M. Schooler, David Zage, Hassnaa Moustafa, and Jeff Sedayao. "A Marauder's Map for the IoT edge". In: *2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC)*. 2019, pp. 236–245. DOI: 10.1109/CIC48465.2019.00038.

[142] François Portet, Michel Vacher, Caroline Golanski, Camille Roux, and Brigitte Meillon. "Design and evaluation of a smart home voice interface for the elderly: acceptability and objection aspects". In: *Personal and Ubiquitous Computing* 17.1 (2013), pp. 127–144. DOI: 10.1007/s00779-011-0470-5.

[143] Eugene Cho, Maria D Molina, and Jinping Wang. "The effects of modality, device, and task differences on perceived human likeness of voice-activated virtual assistants". In: *Cyberpsychology, Behavior, and Social Networking* 22.8 (2019), pp. 515–520.

[144] Eugene Cho. "Hey Google, can I ask you something in private?" In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–9.

[145] Snap, Inc. *Snapchat*. 2019. URL: https://www.snapchat.com (visited on 02/15/2019).

[146] Snap, Inc. *SEC Quarterly Report Form 10-Q*. July 2021. URL: https://investor.snap.com/financials/sec-filings/default.aspx.

[147] Pew Research Center. *Social Media Use in 2021*. Washington, D.C., 2021. URL: https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021.

[148] Josh Constine. *Snapchat launches augmented reality developer platform Lens Studio*. 2017. URL: https://techcrunch.com/2017/12/14/snapchat-developer-platform (visited on 07/11/2021).

[149] Evan Spiegel. *Let's Chat*. 2012. URL: https://web.archive.org/web/20210810094618/https://newsroom.snap.com/lets-chat (visited on 08/10/2021).

[150] SmartThings, Inc. *SmartThings*. URL: https://web.archive.org/web/20210727135832/https://www.smartthings.com/ (visited on 08/10/2021).

[151] Amazon.com, Inc. *Echo Dot (3rd Gen) - Smart speaker with Alexa - Charcoal*. URL: https://web.archive.org/web/20210805031908/https://www.amazon.com/Echo-Dot/dp/B07FZ8S74R (visited on 08/10/2021).

[152] Benjamin Kempke, Pat Pannuto, and Prabal Dutta. "Harmonium: Asymmetric, bandstitched UWB for fast, accurate, and robust indoor localization". In: *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2016, pp. 1–12.

[153] Pat Pannuto, Benjamin Kempke, and Prabal Dutta. "Slocalization: Sub-uW ultra wideband backscatter localization". In: *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2018, pp. 242–253.

[154] Giovanni Campagna, Rakesh Ramesh, Silei Xu, Michael Fischer, and Monica S Lam. "Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant". In: *Proceedings of the 26th International Conference on World Wide Web*. 2017, pp. 341–350.

[155] Mark H Burstein and Drew V McDermott. "Issues in the development of human-computer mixed-initiative planning". In: *Advances in Psychology*. Vol. 113. Elsevier, 1996, pp. 285–303.

[156] Eric Horvitz. "Principles of mixed-initiative user interfaces". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1999, pp. 159–166.

[157] David V Keyson, Marc PAJ de Hoogh, Adinda Freudenthal, and Arnold POS Vermeeren. "The Intelligent Thermostat: A mixed-initiative user interface". In: *CHI'00 Extended Abstracts on Human Factors in Computing Systems*. 2000, pp. 59–60.

[158] Rayoung Yang and Mark W Newman. "Learning from a learning thermostat: Lessons for intelligent systems for the home". In: *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2013, pp. 93–102.

[159] Charles Rich and Candace L Sidner. "Adding a collaborative agent to graphical user interfaces". In: *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*. 1996, pp. 21–30.

[160] Douglas B Moran, Adam J Cheyer, Luc E Julia, David L Martin, and Sangkyu Park. "Multimodal user interfaces in the Open Agent Architecture". In: *Knowledge-Based Systems* 10.5 (1998), pp. 295–303.

[161] Mark Maybury. "Intelligent user interfaces: an introduction". In: *Proceedings of the 4th International Conference on Intelligent User Interfaces*. 1998, pp. 3–4.

[162] Gregory Kuhlmann, Peter Stone, Raymond Mooney, and Jude Shavlik. "Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer". In: *The AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*. San Jose, CA. 2004.

[163] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. "Walk the talk: Connecting language, knowledge, and action in route instructions". In: *Def* 2.6 (2006), p. 4.

[164] Dipendra Misra, John Langford, and Yoav Artzi. "Mapping instructions and visual observations to actions with reinforcement learning". In: *arXiv preprint arXiv:1704.08795* (2017).

[165] Jelena Luketina, Nantas Nardelli, Gregory Farquhar, Jakob Foerster, Jacob Andreas, Edward Grefenstette, Shimon Whiteson, and Tim Rocktäschel. "A survey of reinforcement learning informed by natural language". In: *arXiv preprint arXiv:1906.03926* (2019).

[166] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. "Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6629–6638.

[167] Prasoon Goyal, Scott Niekum, and Raymond J Mooney. "Using natural language for reward shaping in reinforcement learning". In: *arXiv preprint arXiv:1903.02020* (2019).

[168] Henry Lieberman and José Espinosa. "A goal-oriented interface to consumer electronics using planning and commonsense reasoning". In: *Knowledge-Based Systems* 20.6 (2007), pp. 592–606.

[169] Byron Reeves and Clifford Nass. *The Media Equation: How people treat computers, television, and new media like real people*. Cambridge university press Cambridge, United Kingdom, 1996.

[170] Erik Davis. *Techgnosis: Myth, magic & mysticism in the age of information*. North Atlantic Books, 2015.

[171] Tanya N Beran, Alejandro Ramirez-Serrano, Roman Kuzyk, Meghann Fior, and Sarah Nugent. "Understanding how children understand robots: Perceived animism in child–robot interaction". In: *International Journal of Human-Computer Studies* 69.7-8 (2011), pp. 539–550.

[172] Nurit Bird-David. "'Animism' revisited: personhood, environment, and relational epistemology". In: *Current Anthropology* 40.S1 (1999), S67–S91.

[173] Betti Marenko. "Object-relics and their effects: for a neo-animist paradigm". In: *Objets et Communication*. Médiation et Information 30. Editions l'Harmattan, 2010.

[174] Philip Van Allen, Joshua McVeigh-Schultz, Brooklyn Brown, Hye Mi Kim, and Daniel Lara. "AniThings: Animism and heterogeneous multiplicity". In: *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. 2013, pp. 2247–2256.

[175] Betti Marenko. "Neo-animism and design: a new paradigm in object theory". In: *Design and Culture* 6.2 (2014), pp. 219–241.

[176] Betti Marenko and Philip Van Allen. "Animistic design: How to reimagine digital interaction between the human and the nonhuman". In: *Digital Creativity* 27.1 (2016), pp. 52–70.

[177] Jiyoung Ko. "Designing with animism". MA thesis. Carnegie Mellon University, 2017.

[178] Arjun Rajendran Menon. "Animism and anthropomorphism in living spaces: Designing for 'life' in spatial interactions". MA thesis. KTH Royal Institute of Technology, 2020.

[179] Justin Huang and Maya Cakmak. "Supporting mental model accuracy in trigger-action programming". In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 2015, pp. 215–225.

[180] Svetlana Yarosh and Pamela Zave. "Locked or not? Mental models of IoT feature interaction". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 2993–2997.

[181] Valerie Zhao, Lefan Zhang, Bo Wang, Michael L Littman, Shan Lu, and Blase Ur. "Understanding trigger-action programs through novel visualizations of program differences". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–17.

[182] Andrew J Ko and Brad A Myers. "Designing the Whyline: A debugging interface for asking questions about program behavior". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2004, pp. 151–158.

[183] Brian Y Lim, Anind K Dey, and Daniel Avrahami. "Why and why not explanations improve the intelligibility of context-aware intelligent systems". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2009, pp. 2119–2128.

[184] Meghan Clark, Mark W Newman, and Prabal Dutta. "The Big House Dataset: Desired applications and interactions". In: *Proceedings of the First Workshop on Data Acquisition To Analysis*. 2018, pp. 19–20.

Appendix A

Interaction Framings Dataset

The dataset used in the interaction framings analysis in Chapter 3 is available as a supplemental download with the paper "Devices and Data and Agents, Oh My" on the ACM library [63], which can be found at the archival link: https://dl.acm.org/doi/10.1145/3132031

The supplemental material includes the dataset in a database, a detailed description of the database schema, and example Python scripts for accessing the database. The dataset is further described in the paper "The Big House Dataset: Desired Applications and Interactions" [184].