A HUMAN-CENTRIC APPROACH TO AUTONOMOUS ROBOT FAILURES

BY

DANIEL J. BROOKS
B.S. WEST VIRGINIA UNIVERSITY (2009)
M.S. UNIVERSITY OF MASSACHUSETTS LOWELL (2013)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY
COMPUTER SCIENCE
UNIVERSITY OF MASSACHUSETTS LOWELL

Author: ..... Date: 20 April, 2017

Dissertation Chair: .....
Dr. Holly A. Yanco

Committee Member: .....
Dr. Jill Drury

Committee Member: .....
Dr. Aaron Steinfeld

A HUMAN-CENTRIC APPROACH TO AUTONOMOUS ROBOT FAILURES

Daniel J. Brooks

20 April, 2017

Abstract

When robots fail, the consequences can outweigh the system's perceived utility and convenience, possibly even leading to the system's abandonment. These failures are not necessarily caused by a catastrophic breakdown of underlying technologies. For example, a robot performing a functionally useful or necessary behavior that is perceived as inexplicable or unpredictable can have an insidious effect on people's situation awareness and lead to negative user experiences. As autonomous robots transition from a few systems in isolated environments to larger numbers in increasingly public settings, these problems will affect not only the robots' users but also other people who simply happen to be nearby, such as bystanders. What we need are failure-ready robots: robots which may not always work properly, but which are designed to minimize the impact of their failures and shortcomings. While other works have focused on low-level error-detection, redundancy, and other well studied techniques for preventing failures, this thesis takes a human-centric approach to investigating failures and concentrates on people, their goals, and their expectations of the robot. We investigate how people react to varying conditions surrounding failures, look at improving people's situation awareness of autonomous systems, and propose an interaction method that allows untrained people such as bystanders to communicate with, and even aid, nearby robots.

Acknowledgments

I would like to thank my committee members Dr. Holly Yanco, Dr. Aaron Steinfeld, and Dr. Jill Drury for their guidance, encouragement, and patiences throughout this process. Your advice and feedback have made me a better researcher. To Holly, thank you for your tireless work and all the opportunities you have given me over the last 8 years. Your advice, ideas, and feedback have not only helped shape this dissertation, but also the way I do research. Thank you for providing not just myself but everyone who works for you with all the resources we could ever wish for, and for the unwavering support and protection you grant us.

The work presented in this dissertation would not have been possible without the significant help and contributions of Dalton Curtin, Josh Rodriguez, Chris Munroe, James Kuczynski, Chuta Sano, and Momotaz Begum. They, along with my committee, are represented by the "we" used throughout this document.

I would also like to thank all the former and current members of the Robotics Lab and NERVE Center with whom I have had the privilege to work. In particular, I would like to thank Kate Tsui, Munjal Desai, Jim Dalphond, Abe Shultz, Misha Medvedev, Adam Norton, Eric McCann, Mark Micire, Jordan Allspaw, and Carlos Ibarra Lopez for your friendship and everything you have taught me over the years.

Thank you to my entire family and in particular my mother, Lori, for your love, wisdom, and support. To my fiancée, Vicki Crosson, I can't imagine having done this without your love, patience, understanding, help, and encouragement every step of the way.

Finally, thank you to Dr. Alice Frye for your time and advice on statistical methods. This work was supported in part by the National Science Foundation (IIS-1552228).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Fully autonomous robots are progressively becoming capable of operating in the unstructured environments of everyday life. To date, very few fully autonomous robots (with the notable exception of iRobot's Roomba) have been deployed outside of the industrial sector, where people's access to such robots is usually restricted during operation to prevent injuries. In contrast, robots such as Rethink Robotics' Baxter, Aethon's TUG, and Google's self driving cars are designed to operate in the presence of people and interact with them. The expanding presence of robotic services will dramatically increase the occurrences of non-expert human-robot interactions as self-driving cars, delivery drones, robot vacuums, and more become integrated into society. By operating in public spaces, these robots will subject a large number of people to interacting with them on a regular basis, despite many of them not being end users of these systems or even receiving any benefits from their presence.

1.1 Problem Statement / Motivation

This impending proliferation of autonomous robots raises questions about what happens when they malfunction or otherwise interfere with people's daily lives. Our ability to build dependable systems is constantly improving thanks to research in sensor technology, artificial intelligence, error detection, fault tolerant software architectures, and fault prevention [Lussier, Chatila, Ingrand, Killijian, and Powell, 2004; Payton, Keirsey, Kimble, Jimmy, and Rosenblatt, 1992]. However, even the most reliable of these systems will not be immune to occasional failures, and the manner in which they fail can seriously effect users' perception of those systems and the services they provide. Furthermore, robots can fail socially even when the technology powering them is functioning soundly. Despite this, relatively little work has focused on investigating human-robot interactions involving failure. For example, it is still unknown whether people can reliably tell if a robot is operating properly or failing, how they will react or behave after encountering a failing robot, or what can be done to mitigate feelings of frustration, anxiety, anger, disgust, or resentment that might result. What we do know is studies indicate that failure by a robotic service make the robot seem less capable, lowers users' trust, and can make people reluctant to use the service again [Cha, Dragan, and Srinivasa, 2015; Desai, Kaniarasu, Medvedev, Steinfeld, and Yanco, 2013; Lee, Kiesler, Forlizzi, Srinivasa, and Rybski, 2010].

The ramifications of integrating large numbers of fully autonomous machines and artificially intelligent agents into mainstream society needs to be considered as well. In addition to the difficult legal and ethical issues currently being debated, there remain very basic human-robot interaction problems that have yet to be addressed. For example, it can be extremely difficult, if not impossible, to tell if an

autonomous robot is working properly by simply looking at it – even for the people who created it. Nevertheless, people will increasingly find themselves in situations as bystanders to robots, requiring them to make critical decisions about interacting with systems with which they are completely unfamiliar. Especially in inevitable situations where a robot will have broken down or failed, it is easy to imagine people asking themselves questions such as "What is this thing doing?", "Is there something wrong with it?", "Does someone need to know that this machine is just sitting here?", and "Is it safe for me to get close enough to it that I can go around it?"

To our knowledge, there are currently no ubiquitous standards or regulations governing how publicly deployed autonomous systems should minimally communicate with people around them. Instead, all human-robot interactions are currently left up to the designers of each system to decide what, if anything, constitutes appropriate human-robot interaction. Claiming that different robots will naturally require different interaction methods is an appealing dismissal of the issue. After all, it seems reasonable to believe that interacting with a self driving car should somehow be different from an industrial cleaning robot. This is certainly rational from an end user or operator's perspective, whose goals are aligned with the service the robot is providing. However, this approach places a burden on bystanders to figure out how to communicate with or interpret information from each new machine they encounter in order to obtain even the most basic information, such as whether a robot is on or not.

We believe that for these robots to be fully accepted in society, they need to be able to bidirectionally communicate at a basic level with those around them, including being able to convey important information about their operational state to people nearby. But how should these things be accomplished? With regards to conveying operational state information, one possibility is to leverage domain-

specific paradigms, such as having a self-driving car turn on hazard lights to signal if it is having trouble. However, domain-specific paradigms may not translate well to other applications, and paradigms for signaling other types of information may not exist. For example, a tow truck operator may want to know that a self driving car is disabled before trying to move it, but no common indicator currently exists to communicate this information. An office worker may see a courier robot sitting idle in the middle of a busy hallway and wonder if the robot is working properly by waiting for people to clear out before trying to move, has been disabled and can be safely pushed out of the way, or is experiencing problems and someone needs to be notified. Furthermore, specifying different standards of signaling critical information for different domains could quickly become frustrating and confusing.

Communication between between people and robots is an ongoing area of active research. Unfortunately, existing literature on the subject offers little in the way of potentially viable solutions which could be implemented in the near future to establish widespread bidirectional communication between people and autonomous robots. Much of the work in human-robot interaction is frequently focused on a specific robot interacting with end users, and in many cases the research is based on particular attributes or capabilities of the system involved, such as gesturing with mechanical arms, using digitally rendered eyes to indicate the direction of communication, or leveraging techniques rooted in context/domain specific situations. Interactions with commercial robotic products is frequently limited to a single end user and often involves installing specialized software or creating a user account as a prerequisite. While such steps may be a relatively low barrier for someone who wants to make use of the robot and its services, they could be a source of frustration for non-users with little interest in the robot aside from resolving the matter at hand as rapidly as possible.

1.2 Research Contributions

Our goal in this research is to investigate human-robot interactions involving autonomous robotic service failures. We take a human-centric approach to our research, looking at how people react when robotic services fail and what can be done to improve these inevitable situations. We also investigate simple communication paradigms which could potentially be standardized and used across many different kinds of robots to improve human-robot interaction in such situations. Specifically, the contributions of this thesis to our broader research goals include:

- A method for measuring differences in people's reactions to failure scenarios.
- Analysis of how failure severity, context risk, and different types of recovery strategies influence people after a robotic service failure.
- An icon-based communication paradigm which can be used by autonomous robots to provide bystanders with basic levels of situation awareness.
- A smartphone-based human-robot interaction system, intended for use as a ubiquitous secondary interface for enabling minimal or emergency communication with bystanders and untrained people.
- Design principles for the creation of failure-ready robots.

Our work is divided into three components: understanding the effects a failure has on people's perceptions of robotic service failures, improving people's situation awareness around autonomous robots, and devising an interaction method that allows untrained people (such as bystanders) and arbitrary robotic platforms to communicate with each other.

How people react to robot failures and recovery strategies is not yet well understood from a human-robot interaction perspective. Chapter 3 discusses work in

which we conducted two online surveys with a total of 1200 participants who were asked to assess situations where an autonomous robot experienced different kinds of failure. This information was used to construct a measurement scale of people's reaction to failure where positive values correspond with increasingly positive reactions and negative values with negative reactions. We then used this scale to compare different kinds of failure situations, including the severity of the failures, the context risk involved, and the effectiveness of different kinds of recovery strategies. We found evidence that the effectiveness of recovery strategies depends on the task, context, and severity of failure.

When an autonomous robot breaks down, needs help, or is a hazard to people, it needs to be capable of communicating information about its internal state to bystanders who know nothing about its intended function. Such critical information should be quickly and easily obtainable, be devoid of prerequisites (such as requiring access to the internet or a priori knowledge about the system), and be readily understood by people who have not been trained on the robot's use. In Chapter 4, we investigate the use of non-verbal communication through status icons. A series of online surveys were conducted to construct five representative icons and validate their ability to convey information across a wide range of robotic platforms.

We also investigated a smartphone-based communication system which could be used to create a ubiquitous secondary communication channel between people and autonomous robots. The rise in smartphone popularity and their constant presence in society make them an appealing target for the widespread deployment of a new technology. Additionally, smartphones provide a rich media interface, come with a variety of communication modes (e.g. 4G internet, WiFi, Bluetooth, NFC, cameras, etc), and offer a set of previous established interaction paradigms (such as offering location based services and attracting an individual's attention).

Chapter 5 describes an interaction method we have designed that combines these attributes into a simple system intended for untrained people to interact at a basic level with a wide variety of different robots in a uniform format. We discuss our implementation of this system and an experiment we conducted to verify the feasibility of this interaction method.

Finally, we discuss how human-computer interaction (HCI) design guidelines are applicable and relevant to the design of interaction methods for robots being deployed into public settings with a focus on systems that may be experiencing failures or are otherwise causing problems for the people around them. We introduce a set of design principles for failure-ready robots based on existing HCI guidelines, our research, and patterns found in literature.

Chapter 2

Background

2.1 Characterizing Failures

One of the problems encountered when trying to discuss or reason about the topic of failure is the overuse of the term *failure* and generally referring to things as having "failed." Merriam-Webster defines the term failure as "a state of inability to perform a normal function," or alternatively, "a lack of success." However, more precise terminology is necessary to discuss the cause and effect nature of failures.

2.1.1 Defining Failure

Originally developed by the US military, Failure Mode and Effects Analysis (FMEA) defines terminology that is often adopted by areas related to reliability engineering due to its ability to characterize many different aspects surrounding a failure. For example, a *failure* is defined as functionality that is lost (example: no communication with laser device). The *failure mode* is the way the failure occurs. It describes the end state, or the result of the failure cause/mechanism (example: loss of laser hardware device connection to the operating system). The *failure mechanism* is

the underlying reason for the failure (example: power wires to laser hardware were severed). A *failure effect* is the immediate consequences or status resulting from the failure (example: no data received from the laser's device driver). Failure effects can be further classified as either a *local effect*, which is the effect to the component (example: possible damage to printed circuit board (PCB) solder point from lack of strain relief), or an *end effect*, which is the failure effect at the highest level of the system (example: absence of localization or obstacle avoidance behaviors).

We have adopted the use of the following terminology for describing failure (definitions which are consistent with those in Lussier et al. [2004]). A *failure* refers to a degraded state of ability which causes the behavior or service being performed by the system to deviate from the ideal, normal, or correct functionality. A system can experience many different kinds of failures, also known as the system's "failure modes." Such events are often first characterized by their symptoms (also called "failure effects") and their severity.

Failures occur as a result of one or more *errors*, also known as "failure mechanisms," in the mechanical, electrical, or logical (software) components of a robotic system. Some errors can be handled gracefully by the system, thus preventing the occurrence of a failure. Errors can also develop slowly over time (such as sensor drift) rather than being linked to a single event [Payton et al., 1992]. Errors are caused by one or more *faults*, which generally fall into three categories: *physical faults*, *design faults*, and *interaction faults*. Physical faults are caused by "adverse physical phenomena," design faults by unintentional characteristics of the system created during development, and interaction faults from they system's experience in the world. To illustrate how *failures*, *errors*, and *faults* are related, a mobile robot may experience a localization *failure* as a result of *errors* in LIDAR measurements, due to the *fault* of mud covering the sensor after driving through a muddy field.

2.1.2 Failure Classifications

It is unrealistic to assume that we will ever be capable of identifying and individually addressing every problem that could possibly arise in a fully autonomous robot. However, many potential problems are actually quite similar to each other and creating groups or classifications has been previously shown to be useful in identifying appropriate responses [Murphy and Hershberger, 1999]. To date, there is very little existing work that focuses on categorizing the ways in which autonomous systems fail and most of that work is centered on documenting technical faults.

Steinbauer [2013] surveyed teams that participated in various competitions during RoboCup regarding the failure mechanisms, modes, and effects that their team experienced. They also collected information about the frequency of these events and how they impacted performance, as well as measures the team took to deal with the faults. The survey asked teams to categorize failures into a taxonomy proposed by Carlson and Murphy [2005]. The taxonomy had four high level categories, *Interaction, Algorithms/Methods, Software Design/Implementation,* and *Hardware.* These where then broken down further into sub-categories. *Interaction* was divided into *Humans, Agents and Robots*, or the *Environment. Algorithms/Methods* was divided into *Decision Making, Behavior Execution, Perception,* or *Localization and Mapping. Software Design/Implementation* was divided into *Decision Making, Behavior Execution, Perception,* or *Low level. Hardware* was divided into *Platform, Sensors, Manipulators,* or *Controller.* Finally, the survey asked that faults be categorized by how they impacted performance as *non critical, repairable/compensable* (can be repaired during mission), or *terminal* (lead to termination of mission by robot) and their frequency as *never, sporadic, regularly,* or *frequently* (occurs every mission).

Verma [2001] documented a number of robot failures related to hardware, software, and environmental problems, and offered insights into their findings. For example, she found multiple instances in which a problem occurred due sensor failures that could have been avoided had sensor data been used in aggregate. There were also several cases when robots were inadequately prepared for operating in the environment in which they were placed. A robot with large vertical solar panels was blown over by wind, a robot designed to climb down steep slopes fell over when one of its legs failed to contact the ground on a particularly steep slope, and the lens of a camera fogged up in due to temperature differences. Finally, the author found that one group who was working on fault detection ran into difficulty in testing the system due to unwillingness to allow the system to be damaged.

2.1.3 Software Failures

Many large software systems are comprised of multiple independent processes which communicate with each other using middleware such as the Robot Operating System (ROS) [Quigley, Conley, Gerkey, Faust, Foote, Leibs, Wheeler, and Ng, 2009]. In such systems, there are four major types of failures which can occur that are related to the data being passed between modules [Lutz and Woodhouse, 1999]. First, data could be *missing*. This could come in the form of incomplete messages or dropped packets. Second, data could be *incorrect*. This could either be due to data that was originally correct being mangled during transmission, or because the sender generated the data incorrectly to begin with. Next, the *timing* of the data could be bad. Information may be delayed in being sent, creating an illusion of intentional silence. Data could also be sent early, such as before the intended recipient is ready to receive it. Finally, there could be *extra data*. This could be data that was sent multiple times for redundancy but only expected once. It could

also come in the form of a message larger then the receiver expects to handle, thus overflowing a buffer.

Another form of software failure has to do with its execution. Lutz and Woodhouse [1999] outlined four types of events that could occur during processing. First, a process could halt or *terminate abnormally*. This could happen for example as a result an unhandled exception, segmentation fault (perhaps due to a memory leak), or dead-lock. Second, a particular event that was expected to occur *may never happen*. This could happen in the form of a callback or interrupt which never fires, or an if statement that is not triggered. Next, there could be *incorrect logic* in the form of bad assumptions or unforeseen conditions. Finally, there could be problems with *timing or ordering*. Events might take place in a different order then expected, or a waiting period may time-out before information arrives.

2.2 Human-Robot Interaction

Human-robot interaction (HRI) is a specialized form of Human-computer interaction (HCI) that focuses on the study of interactions between humans and robots (e.g. [Yanco and Drury, 2002]). HRI research studies the design of robot control systems and interactions between people and robots, including user interfaces, how people interact with autonomous systems (such as those using artificial intelligence and natural language processing techniques), and investigating the incorporation of robots into society.

2.2.1 HRI Roles

In HRI, *roles* represent various ways in which people interact with robots. In 2003, Jean Scholtz proposed a number of interaction models based on Norman's seven

stages of action [Norman, 2013], which were called *HRI Roles* [Scholtz, 2003]. These roles included "supervisor," "operator," "mechanic," "peer," and "bystander" interactions. Supervisor interactions were characterized as monitoring and controlling an overall situation - being able to specify a particular action to be carried out, or modify long term plans. Operator interactions were focused on actions, and being able to modify software to adjust unacceptable behaviors. Mechanic interactions were similar to those of an operator, except they were physical in nature and relied upon software for testing modifications. Peer interactions were considered to be "teammates" of the robot, people who shared the same goals as the robot. Finally, bystander interactions were characterized by not being able to affect the robot's goals or intentions, but possibly being able to affect the robot's actions by their presence.

These terms were refined a few years later by Yanco and Drury [2004] to reflect progress made in the field. In the updated classifications, *supervisors* are humans who monitor the behaviors of one or more robots and issuing high level goals to be carried out. *Operators* have the ability to control a robot's behavior, either by issuing it actions to perform or via teleoperation. *Mechanics/Programmers* were able to physically alter a robot's hardware or software. *Bystanders* did not have any direct control of the robot, other then as a side effect of their physical presence.

Unfortunately, these terms may no longer fully encompass modern interactions between people and fully autonomous robotic services. For example, many people now own robotic vacuum cleaners which they use to clean their homes. These systems allow people some level of control over their highest level behaviors (e.g. clean the house), such as the ability to manually start or stop the robot or set a schedule for the robot to repeat the behavior on a regular basis (either via an onboard interface, IR remotes, or smartphone apps), but do not offer low level

control one would normally associate with the role an *operator*. On the other hand, despite being able to issue high level goals to be carried out like a *supervisor*, people are unlikely to constantly monitor the robot and cannot be relied up to immediately intervene if problems occur. The same can be said of self-driving cars. The passengers of a self-driving cars may be considered to be operating it if they specified the destination, despite the fact they are not really in control of the vehicle (e.g. the passengers could be asleep). Another example would be the relationship between the driver of a regular car and a self-driving vehicle on the same road. The human driver in the regular car could be classified as a *bystander* since they have no direct control over the robotic vehicle they share the road with. Since they are also being forced to work alongside the robot, this might be considered a *teammate* relationship. However, the goal of the human driver is very likely to be different from the goal of the robot, as they each have their own destinations to which they are trying to navigate.

In the remainder of this work, we use the term "operator" to refer to a robotic service's end user, or the person for whom the robot is performing a task. We expand the term bystander to refer to people who are likely unfamiliar with the system and its behaviors. We also use the term bystander to refer to people who are not the primary users of a system, but nonetheless are stakeholders in the sense that their lives are affected by their proximity to the system.

2.2.2 Situation Awareness

One of the fundamental problems often found underlying human-robot interaction research is providing users with situation awareness that will allow them to make good decisions. Situation Awareness (or SA) is defined by Endsley as *"the perception of the elements in the environment within a volume of time and space, the*

comprehension of their meaning, and the projection of their status in the near future" [Endsley, 1988]. There are three levels of SA, each of which is a prerequisite to being able to obtain the next level:

- Level 1 – perception of the elements in the environment
- Level 2 – comprehension of the current situation
- Level 3 – projection of future status

The first level of SA, *perception*, is the intake of information using sight, sound, touch, or other senses. Level 2 SA, *comprehension*, is understanding the information and making sense of what it means relative to a given context. Finally, Level 3 SA, *projection*, is the ability to predict or forecast future events - a critical skill required to make good decisions.

There are many factors which affect a person's SA. Some of these factors are related to an individual, including their goals, preconceptions, expectations, training and experience, and actual abilities. Other factors relate to particular situations, such as the state of the environment or a person's workload and stress level. Still other factors such as level of automation, interface design, and system capabilities are related to the system itself.

Understanding the role SA plays in how people make decisions and take actions can provide clues into appropriate responses to failure. Norman [2013] describes a seven stage cycle of how people execute actions. The cycle beings with forming a goal, and then progresses through planning the action, specifying a sequence, performing the sequence, perceiving the state of the world, interpreting that information, and comparing the outcome with the original goal. The last stages of the cycle mimic the acquisition of each of the three levels of SA, with level 3 SA being used as the feedback for altering future plans.

2.2.3 Working with Autonomous Robots

Robot automation can be viewed as a spectrum of varying capabilities defining who is making decisions and how those decisions are being carried out. Endsley and Kaber [1999] defined 10 levels of automation based on earlier work by Sheridan and Verplank [1978]. On one end of the spectrum is *manual control*, in which the human is fully in control of both the decision making process and controlling the robot's behaviors. An example of manual control would be a person operating a remote control car. At the other end is *full automation*, where the computer is fully in control of making decisions and carrying out its own actions. In between lies a variety of configurations in which the tasks of system monitoring, generating lists of possible actions, making decisions, and controlling the robot's physical behavior are variously performed by either the human, the robot, or both. For example, a quadcopter would be said to have more autonomy then a remote control car if it automatically holds an altitude and attitude when not receiving input, despite the fact the both vehicles' position and orientation are manually controlled by a human operator. As a system's autonomy increases, the time the system can run productively while being ignored by the operator (known as neglect-time) also increases [Olsen and Wood, 2004]. This decreases the amount of attention the operator needs to give the system at any given moment until ultimately the system is considered unsupervised.

Some autonomous robots are designed to work with people as teammates, collaborators, or partners [Cha, Mataric, and Fong, 2016]. A research survey in 2005 indicated that people wanted robots to be able to act as personal assistants able to perform chores rather then as companions [Dautenhahn, Woods, Kaouri, Walters, Koay, and Werry, 2005], imagining highly predictable systems which could be controlled by talking to them the same way you would talk to a person. Another study

found that people believed robots to be best suited for jobs that require memorization, having keen perception, or which are highly service oriented [Takayama, Ju, and Nass, 2008]. Knepper, Tellex, Li, Roy, and Rus [2015] created robots which can help people assemble furniture. Autonomous robots have even been created to play board games against human opponents [Matuszek, Mayton, Aimi, Deisenroth, Bo, Chu, Kung, LeGrand, Smith, and Fox, 2011; Brooks, McCann, Allspaw, Medvedev, and Yanco, 2015]. Finally, some autonomous robots are designed to provide services to humans, such as robotic vacuum cleaners or delivery robots.

Control of autonomous robots often takes the form of very high level directions, such as specifying a desired end state and pressing a "start button," demonstrating a task to be repeated (i.e. learning from demonstration [Argall, Chernova, Veloso, and Browning, 2009]), or giving instructions using natural language [Matuszek, Herbst, Zettlemoyer, and Fox, 2013; Brooks, Lignos, Finucane, Medvedev, Perera, Raman, Kress-Gazit, Marcus, and Yanco, 2012]. Unfortunately, abstracting the control of a system by increasing its autonomy can introduce problems related to people's understanding of the system they are using. An excellent example of this is the "Out-of-the-loop Problem" as discussed by Kaber and Endsley [1997]. When a human operator is absent from a system's control loop, they lack situation awareness concerning the system's state. Although this does not pose any problems while the system is operating normally, when the system fails the human operator may be slow or possibly even unable to identify the cause of problems. Even if the operator is able to correctly diagnose the problem, they may not know what steps should be taken to correct the situation. Worse, the operator might not even have the skill set necessary to perform the corrective actions due to their lack of regular involvement in manually controlling the system.

2.2.4 Conveying Information

The ability to effectively communicate information between people and robots is an important area of research in HRI, as it is fundamental to maintaining people's SA. As such, a significant amount of work has been performed investigating how robots can communicate information to people.

Communicating information from manually controlled robots is often very different from autonomous systems. Research with manually controlled robots often focuses on remote teleoperation interfaces. These interfaces commonly provide information to the operator using visual interfaces in the form of live video feeds, sensor measurement displays such as proximity attitude measurements, system health (power, communication, warning signals, etc), and feedback from specialized sensor payloads. Other modes of operator feedback include audio such as warning noises/cues or sounds from the remote environment, and haptic feedback such as vibrations or force feedback joysticks. Because it can be difficult for a single person to keep track of so much information, techniques such as sensor fusion and display overlays can be used to help operators perceive and comprehend the robot's state and remote environment [Baker, Casey, Keyes, and Yanco, 2004].

Autonomous robots, on the other hand, often require different methods and modalities than traditional interfaces for communicating with people. Frequently, the subject of the information being communicated has to do with the system's internal state or mission status. This communication is sometimes focused on the robot's operator, or in the case of fully autonomous robots, the system's users, while in other cases it is designed for use by bystanders.

One common method of visually conveying state information is the use of external lights. Baraka, Rosenthal, and Veloso [2016] experimented with using expressive lights to communicate information about an autonomous service robot's state

to people. The robot used animated colored lights to provide information about task progress, that the robot's movement was being obstructed, and that the robot required a human to intervene. In their study, participants were more accurately able to answer questions about why a robot shown to them in a video behaved a certain way when the lights were being used. An LED strip was used by Szafir, Mutlu, and Fong [2015] to create a light ring to indicate the intended direction of flight by a quadcopter to the people around it. Various illumination patterns representing several different paradigms (e.g. blinking turning indicators, thrusters, etc) were tested to determine how quickly and accurately each communicated the robot's intent. Bethel and Murphy [2008] investigated the use of a robot's movements, ambient light colors, non-verbal sounds, and proximity to people to convey affective expression in socially acceptable ways.

Light is not the only visual method of communicating information about a robot's state. Non-verbal behaviors and facial gestures were used by a robot to provide feedback about the robot's level of comprehension after a person demonstrated a task to it [Breazeal, Kidd, Thomaz, Hoffman, and Berlin, 2005]. Text-based communication (similar to text messaging) was used by an autonomous system to send status updates to a remote operator as it carried out commands [Brooks et al., 2012].

The use of signs and symbols to convey information is one of the oldest forms of communication; the study of signs and their meanings is known as Semiotics. Signs may have varying interpretations across different cultures, and their meanings may change or even be lost over time [Frutiger, 1989]. Bowie and Bowie [2009] performed a study based on 10 road signs from the United States, presenting participants with 88 variations in order to determine the characteristics (such as shape, color, and symbols) that most effectively conveyed meaning to drivers. They found that

while some signs (such as the crosswalk) were very effective (100% recognition), others (such as the school warning and school crosswalk) failed to convey their meaning to over half the participants. In most (but not all) instances, signs with correct wording printed on them on them were no more effective then the same sign without words. However, when incorrect or conflicting wording was substituted, even easily recognizable signs such as the stop sign (which had 100% recognition unmodified) were found to be interpreted differently. Icons have been used to indicate an autonomous robot's confidence level in its own ability to carry out a navigation task [Desai et al., 2013]. Both semantic (smiling/frowning faces) and non-semantic (plus and minus symbols) icons were tested, and semantic indicators were found to cause more sudden changes to users' trust in the system.

Audio is another method of communicating information about the status of a system. Speech has been employed to allow robots to ask people for help, either as an attempt to recover from a failure or as part of a routine task. Work in the area of natural language communication has leveraged inverse semantics to generate tailored dialog for communicating a request for help to a person [Knepper et al., 2015]. Hüttenrauch and Severinson Eklundh [2006] performed an experiment in which a robot without manipulators verbally asked a bystander to help it retrieve a cup of coffee to deliver.

Fischer, Soto, Pantofaru, and Takayama [2014] investigated the use of gestures and different kinds of audio signaling (verbal greeting vs. an acoustic beep) to attract a person's attention and make a request for help. They found that the verbal greeting seemed to be the most reliable way of attracting attention, but it did not improve the likelihood of the person to perform the robot's request. Cha et al. [2016] used a combination of both sound and light to indicate to a nearby person that it needed help with varying degrees of urgency. Their preliminary

results suggested that people were using the light and sound in different ways, with the sound initially attracting their attention to the robot and then interpreting the level of urgency of the request by looking at the light.

Gestures have also been combined with other effects to increase a robot's persuasiveness when influencing people's behavior. Chidambaram, Chiang, and Mutlu [2012] found that a robot's proximity to a person, the use of directed gaze (head turning), and arm movement gestures impacted participants' compliance with respect a robot's verbal suggestions during an experiment. Riek, Rabinowitch, Bremner, Pipe, Fraser, and Robinson [2010] tested people's reaction times for the gestures "beckon," "give," and "shake hands" when shown the gestures from different angles (front and side) and using different implementations (smooth vs. abrupt). However, in their experiment participants were first shown training videos of the robot's gestures to help them identify the robot's intended movements later. Gestures have also been used to provide feedback to operators after they give instructions. For example, Hiroi and Ito [2013] built a robot that uses an arm to point to the location to which it thinks the operator wants it to travel.

2.2.4.1 Media Richness Theory

With so many different modalities for communicating information, it is natural to wonder whether some methods are inherently better then others. Media richness theory tries to explain how different forms of communication affect task performance. It proposes that task performance can be improved when "task-information processing requirements" match a medium's "ability to convey information richness," where a media's richness is defined as the media's feedback capabilities and variety of communication channels used. Thus, a face to face interaction would be considered the richest method of communication, while formal written docu-

ments that follow strict protocols (such as numerical spreadsheets) are considered very "lean." According to the theory, rich media allows people to achieve mutual understanding about abstract and complex concepts while leaner media is better for more routine communications. The theory suggests that when a media is too rich for the task being performed it becomes distracting and therefore inefficient, while media that is not rich enough for a given task becomes ineffective due to its inability to transmit the necessary information.

Suh [1999] performed a study to test the effects of how media richness impacts task performance during dyadic communication with respect to the theory's predictions. The independent variables were task characteristics and communication-media characteristics (text, audio, audio/video, and face-to-face) and the dependent variables were task performance (decision quality and decision time) and satisfaction (for process satisfaction and outcome satisfaction). The results of their study, in agreement with other similar studies, did not end up supporting this theory. Instead they found that there was no relationship between task and media pairings on the quality of the interaction, that audio took the least amount of time (regardless of the task), and that there were no strong correlations between the task performance and satisfaction. In other words, the most effective medium was not necessarily the most satisfying.

2.2.5 Trust and Risk

Relying on autonomous robots to perform services means putting a certain amount of trust in the system and taking some risk. Psychologists have found that there is a strong negative correlation between the way people perceive the risk and benefit of activities, despite the fact that in the real world risk and benefit are usually positively correlated [Slovic and Peters, 2006]. In other words, if someone perceives

the benefit of a robotic service as being high, they will infer the risk to be low. Both the inverse and reverse relationships also hold; if a person perceives the benefit to be low, they will infer the risk to be high, and if they perceive the risk to be high, they will infer the benefit to be low. Studies have shown people to make detrimental decisions that reflect this kind of reasoning even while being consciously aware of the flaw in their logic, simply because they "felt" like they had made the right choice [Denes-Raj and Epstein, 1994].

A system's level of reliability has been found to have a strong effect on the operator's trust is that system. Desai et al. [2013] performed experiments in which they artificially manipulated the reliability of an autonomous robot's performance. They found that first impressions mattered, as scenarios in which the robot experienced periods of low reliability early on resulted in lower levels of trust by the operator the remainder of the experiment, while reliability drops later in the interaction were not as detrimental. Moray, Inagaki, and Itoh [2000] showed that operators preserve their own self-confidence in using systems when they can distinguish between their manual actions and the actions of the automation.

Risk can be perceived in two different ways - analytically (using logic and reasoning) and experientially (using feelings, instincts, and intuition) [Slovic and Peters, 2006]. The latter has been credited as the primary influence for motivating people's behaviors due to it being a faster and easier decision-making method for assessing dangers.

This has important implications with respect to robots experiencing failures. If the experience of a failure results in a perceived increase of risk either from using the robot or being in its presence, people will also infer a lower benefit of using the system. On the other hand, if the perception of risk can be suppressed or mitigated in the event of failures, the inferred benefits of using the system

should remain high. Alternatively, the theory suggests that it may be possible for a technology to be perceived as so beneficial that the inferred risk would be low enough for users to overlook or turn a blind eye to system failures (although this seems unlikely). Another possibility is that the risk is perceived as being so low that poor autonomy is trusted, such as in the case of the Roomba. Research in assistive technologies has identified "enhancement of user performance," device effectiveness, and reliability to all be factors in technology abandonment, with researchers' recommending "careful analysis of the costs and benefits of device use from the consumer's perspective" [Riemer-Reiss and Wacker, 2000].

2.3 Identifying and Preventing Causes of Failure

People have always sought to create reliable and robust tools and products as a result of the undesirable consequences that stem from failures. A significant amount of energy and effort is invested in preventing failures in order to reduce or eliminate time delays, additional expenses, unacceptable outcomes, loss of trust, unsafe/hazardous situations, and other harmful side effects.

Lussier et al. [2004] posited the attributes of a dependable system were availability, reliability, safety, confidentiality, integrity, and maintainability with the following definitions. The *availability* of a system is its ability to deliver a (correct) service at a given time. The *reliability* of a system is its ability to continuously deliver the (correct) service over a period of time. A system is *safe* if there is an absence of catastrophic consequences for both the users and the environment. *Confidentiality* is the ability to prevent the unauthorized disclosure of information, while a system's *integrity* is its ability to prevent improper system state alterations. Finally, *maintainability* is the ability for the system to undergo repairs and modifications.

Lussier also divided the techniques used to create dependable systems into four categories: fault prevention, fault removal, fault tolerance, and fault forecasting. The first two categories (prevention and removal) focus on avoiding faults, while tolerance and forecasting focus on accepting and dealing with faults. Fault tolerant systems are usually hallmarked by error detection and recovery.

2.3.1 Reliability Engineering

One of the ways that software engineering differs from other branches of engineering is that reliability can be very difficult to achieve. Physical materials such as building supplies and electronics have well understood properties and they fail due to external conditions being applied to them. Software frequently doesn't usually fail this way; instead it can behave incorrectly as a result of errors in design or implementation. As a result, typical strategies for increasing reliability, such as redundancy, are not necessarily effective with software. For example, to add redundancy to the space shuttle's on-board computers, the backup systems had to be designed with completely different software [Pentti and Atte, 2002; Pecheur, 2000]. This section describes several techniques employed by reliability engineers to create highly robust systems.

2.3.1.1 Failure Mode and Effects Analysis

As previously discussed in Section 2.1.1, Failure Mode and Effects Analysis (FMEA) is a method of systematically analyzing the effect of a component failure within a larger system using a form of inductive reasoning. The goal of an FMEA is to determine and understand the causes of various kinds of failures to identify methods of reducing the probability of such occurrences in the future. This is accomplished by documenting all the failure modes of every component within the system. FMEA

can be used to improve the quality and reliability of a system and identify potential problems before they occur [Pentti and Atte, 2002].

The analysis is performed within the context of a set of predefined assumptions, such as nominal system power is always available. An important shortcoming of FMEA is that it only considers a single source of failure at any time. In other words, the effects of a component's failure in the FMEA are not considered in the context of any other simultaneous failures occurring. This prevents it from being able to identify complex failure modes or predict the likelihood of high level failures. Another shortcoming of FMEAs are that they do not normally consider the effects of external events on the system. Nonetheless, FMEAs are good at identifying all the sources of failures in a system, and identifying their immediate local effects.

2.3.1.2 Fault Tree Analysis

Fault Tree Analysis (FTA) has also been used in software engineering for debugging purposes. While FMEA is usually a bottom-up method of analyzing a system, FTA is a top-down method that focuses on the outcome of certain events by investigating undesirable conditions as a function of the logic and events leading up to that condition. FTA is a form of deductive reasoning. An FTA takes the form of a tree structure that consists of logic gate symbols which represent the relationship between multiple events that can lead to the undesirable condition (represented as the root of the tree). The timespan in which these events can occur is usually defined, such as the duration of a mission. Since a single event can have multiple effects, it is possible for that event to occur in multiple places within the tree, and is called a "common cause." FTA's were originally developed to be able to calculate the probability of a system failure given the probability of individual pieces of hardware failing. FTA is effective at identifying a system's resilience to a par-

ticular event occurring and, unlike FMEA, does take external events into account. However, FTA is not well suited to finding all the possible faults in a system.

2.3.1.3 Software Verification and Validation

Software Verification and Validation (V&V) is a process that checks if a software system meets design requirements and works as it was intended. *Verification* refers to the process of checking that a software implementation has been built according to specifications (e.g. Did we build it correctly according to the plans?), while *Validation* confirms that it meets requirements (e.g. Does it do what we need it to do? Does it do the right thing?).

In addition to potentially being very resource intensive, this is often a very manual and time consuming process. For example, one common approach to verification is called *model checking*, in which a model of the software is systematically searched for violations of specified rules, starting from some initial states and repeatedly transitioning through all reachable states until the entire model has been covered. As the size of the system grows, the number of states needing to be checked can also grow exponentially. However, before this can even happen, the software must be manually translated into an abstract model that is represented using a mathematical language [Pecheur, 2000].

2.3.2 Error and Failure Detection

Hardware replication has been used to improve reliability at the physical level by including redundant actuators and sensors [Kabuka, Harjadi, and Younis, 1990]. Using this strategy, bad sensors can be identified and suppressed by comparing the values of all the redundant sensors against each other and reporting the most common value as the "real value." However, this solution is expensive, impractical

for space-constrained applications, and naively trusts that the majority of sensors are working properly. Additionally, it does not protect the system from changes in the environment which the system is unprepared to interpret. Ferrell [1994] improved upon this idea by using complementary sensors and a priori knowledge to derive expected values for each other.

Becker and Flick [1996] listed a number of ways software failures could be detected. First, they proposed using a hearbeat monitor to detect when a coordinated process fails to perform a particular function. Messages sent between modules can be labeled in a numbered sequence to detect when messages are received out of order or missed entirely. System resources can be monitored to detect when memory limits are being approached or a filesystem is having problems. A process manager can check to ensure that all processes which are supposed to be running are present. Finally, applications can have internal detection systems that monitor buffers and queues, along with explicit error detection code designed to sanity check process state.

Murphy and Hershberger [1999] detected errors using the generate-and-test approach, a technique that makes use of the robot's ability to interact with its surroundings by forming and testing hypotheses about the nature of anomalous sensor data. Categories were manually linked with possible causes (hypotheses) and recovery behaviors into a precomputed library which could be searched. Each hypothesis consisted of a categorization for quickly narrowing down the set of hypothesis that needed to be tested, test methods for evaluating if that hypothesis could be the cause of a problem, a list of relevant sensors that would be affected if the hypothesis were true, and a recovery method. Failures at runtime could then be categorized as a sensor malfunction, an environmental change, or an "errant expectation" by generating a hypotheses as to the cause and performing pre-described tests. Many

hypotheses were concerned with a single physical sensor, while others were designed to detect changes in the environment. Testing could be bypassed in the case that the list of hypotheses could be narrowed down to a set which all point to the same recovery method before testing individual hypotheses begun. In the case that no hypotheses trigger but a problem is still detected, a behavioral failure was assumed.

Canham, Jackson, and Tyrrell [2003] used a technology called an artificial immune system (AIS) to detect errors. AIS is inspired by biological immune system and based on the concept of being able to detect the difference between "self" and "non-self." As the system runs, it learns what "normal" data looks like. The authors used this concept to "immunize" a mobile robot to avoid objects by training the system while never driving it into obstacles. Thus, the robot would later detect instances in which it drove towards a nearby object in a straight line as "non-self."

2.3.3 Failure Handling and Robust Systems

Once errors can successfully be detected in a system, this information can be used to try to make the system more reliable. In this section, we discuss some of the techniques that have been used to create systems which are robust to encountering certain kinds of errors.

2.3.3.1 Planning for Success

For most modern robotic systems, it is not practical (and possibly impossible) to anticipate and handle every possibly way in which the system could fail. Payton et al. [1992] used a task-oriented approach they called "do whatever works" that focused on identifying various ways in which the system could perform successfully, rather then trying to define special case instructions for all the ways things could break. Their system design was derived from subsumption architectures and

implemented a distributed and redundant system of control behaviors along with a mission manager to track and prioritize various mission goals to achieve high level objectives. A behavior became activated if there was a mission goal it could contribute towards and its required sensory inputs were met. Behaviors could also inhibit other behaviors to prevent conflicting actions. The various active behaviors would check to see whether they were being effective at achieving mission goals, with less effective behaviors deferring control to the more effective behaviors. Having redundant behaviors that achieved the same goal in different ways provided the system with fault tolerances while also performing optimally without the need to identify or diagnose specific problems.

Many modern robots that employ autonomy use a variation of a classic three-layer architecture [Gat et al., 1998], which consists of low level and reactive controllers, intermediate level actions or behaviors, and high level goal planners. Ingrand, Chatila, and Alami [2001] described how such architectures could be made more reliable by designing each module to be capable of operating within a variety of established contexts, outside of which the system takes responsibility for maintaining reliability through decision making.

Murphy and Hershberger [1999] used a library of of pre-categorized failures (discussed in Section 2.3.2) to determine an appropriate recovery method to use as a response to a variety of situations. Three types of recovery methods were also specified, including reconfiguration, recalibration, and corrective actions. Reconfiguration consisted of maintaining the current behavior while swapping out the source of a particular piece of sensor data for an equivalent one, or by replacing a behavior that is no longer possible with an alternative behavior. Recalibration of sensors involved actions specific to the devices such as performing a re-alignment on a pan/tilt unit or focusing a camera. Corrective actions attempted to fix problems

using last-ditch attempts, such as shaking a sensor.

The world is full of uncertainties and rapidly evolving situations that can be difficult to model. This means there are times when planned actions will fail due to inaccurate information about the state of the world. Mendoza, Veloso, and Simmons [2015] created a planner for a team of RoboCup robots which not only generated sequences of actions but also expectations about the results of those planned actions that could later be compared with sensing observations once the plan had been executed. This information was then used to identify statistically significant situations in which the existing feature-space model poorly represented likely outcomes, and then used to update the model for future use by the planner. A limitation to this technique is that it requires a priori domain-specific knowledge for generating future expectations.

2.3.3.2 Propagation, Confinement, and Alternative Behaviors

The correct manner in which to respond to a failure may depend on the nature of the failure and its source. In some scenarios, it may be possible and desirable to mask the failure from the rest of the system, allowing the system to continue operating as it is. In other scenarios, it might be necessary to alter the system's behavior to compensate for its inability to continue working at the status quo.

Ferrell [1994] suggests that failures should be confined as early as possible, thus preventing unchecked errors from spreading through the system. The further the error propagates, the more diverse the problems it causes become, which in turn increases the complexity of handling them. In some circumstances, redundant capabilities may be used as a replacement for a malfunctioning part of the system. Other times, it may be possible to adapt the robot's behavior to compensate or mitigate the loss of functionality.

In some cases, it may be possible to alter a robot's behavior to perform in a clearly less desirable manner compared to nominal functionality to avoid a more catastrophic result. Mueller and D'Andrea [2014] described work in which varying degrees of control could be maintained over a quadcopter despite losing up to 3 of its 4 propellers. With 3 or 2 (opposing) functioning propellers, a quadcopter can take off, maintain a target altitude, translate in an intended direction, and land again. This is achieved by allowing the vehicle to rotate about an axis in space, thus abandoning the normally desirable level of control to achieve a less desirable but sustainable level of control. The implications of this technology are that a failing quadcopter could be caused to crash land in a particular place or direction, thus mitigating some of the negative consequences that might otherwise occur.

2.4 Dealing with Failures

The notion that we will never have perfectly reliable robots raises questions about what can be done to mitigate the consequences after a failure occurs, referred to as recovery strategies. Providing users with advanced warnings of potential problems has been shown to improve users' evaluations of a system after a failure, and activities such as offering an apology can sometimes make the robot seem more competent [Lee et al., 2010]. Analysis of real-time user trust with an autonomous robot found that the robot could provide the operator with confidence feedback on its current performance to encourage better control allocation without altering the user's level of trust in the system [Desai et al., 2013]. Researchers have explored having robots seek out nearby people to ask for help [Rosenthal, Veloso, and Dey, 2012]. Work on generating failure-specific natural language requests for help based on the robot's task indicated that users had a more enjoyable experience compared

to more generic methods of requesting help [Knepper et al., 2015].

Unfortunately, not all recovery strategies work the way they are intended. If people are not made aware of why a robot is behaving in a particular way it can lead to confusion. In one case, workers at a hospital were documented blaming each other for having "messed up" an autonomous delivery robot after they observed it behaving inexplicably, while in reality the robot was performing a calibration routine [Kim and Hinds, 2006]. However, providing users with information about the cause of a failure could also make the situation worse. Experiments have shown that users respond very negatively when a robot blames them for causing a failure, compared to when blame was collectively assigned to both the user and robot as a team (e.g. using "we" statements), even in cases where the human was likely aware that they were the primary source of the problem [Groom, Chen, Johnson, Kara, and Nass, 2010]. That said, having the robot blame anyone (even itself) for a failure has been shown to cause users to lose trust in the system [Kaniarasu and Steinfeld, 2014].

Taxonomies have been described by Carlson and Murphy [2005] and Steinbauer [2013] which categorize faults and provide insight into the many complex ways a system could fail. Attributes of a "dependable" system have been described by Lussier et al. [2004] (see Section 2.2.5). Concepts taken from consumer market research have been shown to have analogous effects in robotic services [Lee et al., 2010]. However, to the best of our knowledge there is no theoretical model that characterizes failures of autonomous robots to predict people's reaction to various situations.

2.4.1 Perception of Robot Capabilities

Cha et al. [2015] investigated the relationship between people's perception of a

robot's capabilities and actual robot capabilities. They found that people perceived robots which use generated speech to be more capable (both socially and physically) than those that do not, with robots capable of speaking conversationally perceived as being even more capable then those which were only capable of a functional level of speech. However, they noted that in cases of failure that the robots with a conversational level of speech were perceived as less capable than those with functional speech. The authors proposed one reason for this switch may be due to people having higher expectations for conversational robots which become challenged by the occurrence of failures, while functional speech revealed more about the robot's actual limitations and afforded more realistic expectations.

Desai et al. [2013] looked at how people's trust in robotic system evolves over time with respect to the system's reliability by having participants drive a robot through an obstacle course while varying the reliability of the system at different points in time. Participants could choose between two control allocation strategies - letting the robot drive itself autonomously or manually piloting it themselves. They found that reliability failures early in a person's experience resulted in much lower trust, causing people to prefer suboptimal control strategies. Additionally, they investigated how having the robot provide feedback to the user of its "confidence" in its own abilities affected operator's choice of control allocation. Their results indicated that participants were inclined to believe the feedback indications and would switch control strategies according to the feedback. They also found that false indications of reliability drops by the feedback system did not seem to significantly alter users real-time level of trust in the system.

2.4.2 Communicating Failures

Communication is an important part of any relationship. When autonomous systems experience failures, it is critical that they are capable of communicating information to both the operators or people responsible for the robot and to people who happen to be nearby, since to an observer a robotic system may appear to be working properly if it is outwardly behaving as expected. While some failures may manifest themselves in an obvious manner such as behaving erratically, it could easily be the case that a failure goes unnoticed because of the system being unsupervised or simply being unrecognized due to the poor SA. Bystanders might not be familiar enough with a system to be able to tell whether a robot is working properly, and even operators might not notice signs of trouble due to being out-of-the-loop. Thus, the ability for robots to be able to signal to people when failures occur is an important feature.

Rosenfeld, Agmon, Maksimov, Azaria, and Kraus [2015] created an intelligent advising interface designed to allow a single human operator to control multiple autonomous robots. They built a supervisory interface that consisted of a global map view of multiple robots, a thumbnail view of critical information for each robot, and a large teleoperation interface that could be used to control one robot at a time. The interface also alerted the operator whenever a robot required the person's attention, which occurred whenever the robot encountered a problem or needed human confirmation of an action. For example, if a robot was stuck the interface would alert the operator and advise them that "Robot i is stuck, try to get it loose," and then allow the operator to assume manual control over the machine. To determine when the operator should be contacted, they mapped their task space onto a Markov Decision Process and trained their model in a "utopic" simulation where the robots would not suffer from malfunctions.

Failures in autonomous systems can be communicated implicitly if the robot is able to clearly communicate its intentions in a way that can be contrasted with physical behavior when an error occurs. The use of intuitive visual or audio signals to convey intention could passively provide people with knowledge that a problem existed, and possibly even information as to the cause of a problem. For example, if a drone were equipped with a light ring direction indicator like the ones described in Szafir et al. [2015] and was indicating a straight flight path while the drone was translating to the side, users familiar with the drone's normal operation would be able to immediately discern that something was not right.

Hiroi and Ito [2013] built a robot intended to accompany people through daily life. Recognizing that the robot would not be able to achieve perfect interaction 100% of the time, the authors built in failure mitigation techniques. The user could control the robot by using speech and pointing to a location to tell the robot to move to a specific position. The robot would then point its manipulator arms towards the place it believed the person wanted it to go, allowing the person to know ahead of time where the robot will be going and giving them a chance to intervene. The robot was also capable of tracking and following a person. In the event that the robot lost track of where the person was, it would call out to the user asking them to come back and stand in a location the robot points to so that the robot can find the person again. The concept of asking a person to perform some action to aid the robot when it experiences a problem is discussed further in Section 2.4.4.

2.4.3 Recovery Strategies

Lee et al. [2010] performed a survey study in which different techniques for mitigating failures from a robot were tested. In the experiment, people were presented

with a scenario in which one of two robots (one humanoid and one not) would be summoned by a user to deliver them a can of Coke. For everyone except a control population, the robot would fail at the task by way of delivering a different type of soda. They manipulated multiple variables attempting to mitigate the negative consequences, including whether or not to give the user a expectation that the robot may not work correctly along with various types of "recovery strategies" such as apologizing, compensating the user, or offering different options. They found that setting the level of expectancy so that the user knew the robot might fail was particularly effective in preventing negative reviews, and that having the robot apologize made the robot seem more competent. They also found that it was better to apologize to people who treated the robot more like an agent, while compensation was better for people who treated it like a tool. Furthermore, apologizing and providing options to the user increased the perception that the user would use the service again. These results strongly suggest that the manner in which the system communicates with humans after experiencing a failure will have an influence on their perception of the system.

2.4.4 Asking for Help

One possibility for mitigating a failure is the potential of recovering from certain situations by asking for human intervention. For example, if the electronic mechanism normally used by a robot to open a door were to suddenly stop working, the robot could possibly ask a nearby person to open the door for it. Cha et al. [2016] summarized the process of asking for help as being broken down into three phases: 1) getting someone's attention, 2) indicating to the person that help is needed, and 3) conveying the request for help.

Knepper et al. [2015] used natural language to construct specific requests for help from people which are designed to simultaneously improve their SA. They tested their system in an experiment during which a person helped a robot assemble Ikea furniture. When the robot encountered a problem, the system would generate specific instructions based on the robot's need in a manner that would remove as much ambiguity from the request as possible. Users reported that they felt the system was more effective at communicating needs than other tested methods; however, it did not make them any more proficient at task switching.

In another experiment, participants were left by themselves in a public kitchen area when a robot approached and asked them to pour and place a cup of coffee on it [Hüttenrauch and Severinson Eklundh, 2006]. The experimenters believed that people would help the robot if they understood what the robot wanted, were in a position to be able to help the robot, and knew how to provide the help (given the particular situation). Half of the participants complied with the robot's request and gave the coffee to the robot. The vast majority of the people who helped the robot were those who were not busy concentrating on another task (one of the experimental conditions), while those who were busy responded variously by ignoring the robot, tricking it into thinking they had given it the coffee so it would go away, or by shutting the door to keep the robot out. A little over half the people who helped the robot indicated that by helping the robot they understood they were helping another person (robots don't drink coffee). Giving people a prior introduction to the robot and explaining what it did and how it worked did not have a significant influence on people's willingness to help.

Yasuda and Matsumoto [2013] hypothesized that people may even be able to relate better to imperfect robots that experience failures, viewing them as similar to children or infants who try but fail in their efforts. They experimented with a

robot trashcan that would sometimes spill garbage, but lacking manipulators was unable to clean up after itself. Thus, whenever this would occur, the robot would ask a person to pick up the trash for it followed by "bowing" in the local expression of appreciation. The majority of people found the experience to be positive, even when the robot spilled trash.

Rosenthal et al. [2012] designed CoBots to actively seek out and solicit the help of bystanders in the environment, with the goals of trying to distribute the burden of helping the robot across many different people and anticipating that people will not always be readily available to help and thus must be actively sought. They also explored the idea that people are more likely to help the robot if they believe they will be reciprocated or rewarded. The robot could take messages to people and deliver mail to building occupants, and would sometimes gift people who helped it with candy. People could help the robot with problems such as localization, moving obstacles, and writing notes. Offering gifts of candy did not seem to impact the frequency of people helping the robot.

Asking people for help due to an error or after encountering an unexpected situation could be perceived very differently by users compared to "needy" robots which assume that people can actively be counted on to perform routine tasks that are part of a robot's job. During the CoBot experiments, the authors found that after a few days many people were closing their office doors [Rosenthal et al., 2012]. In a study documenting the use of hospital delivery robots, some of the staff who had to work with the robots ended up resenting them [Mutlu and Forlizzi, 2008]. Instead of the robot's being helpful to the staff, one staff member described the situation as "... more like staff helping the [robot]. I'm the one loading the trays on to it and loading the linen onto it."

2.4.5 Explaining the Causes of Failure

As mentioned previously, one of the challenges of dealing with unsupervised autonomous systems is the out-of-the-loop problem. Thus, it can be expected that even if people know there is a problem, they still won't know the cause and likely won't know the appropriate response to the situation. Not knowing what caused the robot to fail in one instance but not another or why the machine is behaving in a particularly unexpected manner can be very frustrating since it undermines people's confidence in predicting how the robot will perform in the future. The ability to explain why autonomous agents behave in particular ways is currently an active area of research.

Prior work has investigated creating introspective systems capable of explaining why a robot behaved in a particular manner by tracing and logging the flow of information through a system, and keeping track of which pieces of data were used in making progressively higher level decisions [Brooks, Shultz, Desai, Kovac, and Yanco, 2010]. However, providing users with information about the cause of a failure could also make the situation worse. For example, Kim and Hinds [2006] found that robots that attempt to explain their ambiguous actions and errors can actually decrease people's perceived understanding of the system.

Additionally, care must be taken in the manner in which causes of failure are presented to people. While an event can be attributed to a cause, the process of assigning blame involves assessing who was responsible - an important aspect of trying to understand complex situations which often surround failures. Kim and Hinds [2006] noted that people tended to take less responsibility for problems that occurred during a task while an autonomous robot was involved, which can be problematic for trying to provide users with useful feedback regarding failure mechanisms where the user is to blame.

Groom et al. [2010] looked at how people responded to an autonomous robot after it attempted to assign blame for a recent failure. They found that users consistently responded more negatively to robots which blamed them for causing a failure compared to when the robot blamed itself or when blame was collectively assigned to both the user and robot as a team (e.g. using "we" statements). The effect was so strong the authors recommended avoiding explicitly blaming humans in favor of blaming the "team" whenever it is believable and acceptable, even in cases where the human may be aware they were the primary source of the problem. According to the paper *"Only in cases where the source of failure is so obviously attributable to the human that the robot cannot be implicated should human blame be considered."* That said, having the robot blame anyone (even itself) for a failure has been shown to cause users to lose trust in the system [Kaniarasu and Steinfeld, 2014].

Chapter 3

Analysis of Reactions Towards Failures and Recovery Strategies for Autonomous Robots

In Section 2.4.3, we discussed the need for a method to measure people's reactions to robot failures. In this chapter, we demonstrate a method for comparing the detrimental impact of various failures and how effective different types of recovery strategies are at mitigating the resulting negative effects, as perceived by users. We performed a survey experiment looking at different types of failures occurring in various situations. This information was used to construct a measurement scale of people's reaction to failure, which was then used to compare how the severity of the failures, the context risk involved, and the effectiveness of recovery strategies impact people's reactions. For the purposes of this experiment, we grouped recovery strategies into two categories, *task support* and *human support*.

One of the consequences of a failure occurring in a fully autonomous robot system is a deterioration in the task performance, possibly to the point that the

task can no longer be performed. However, an autonomous robot may still be able to take actions that can assist in furthering the task towards completion even in conditions where a failure has rendered the system incapable of carrying it out on its own. We call recovery strategies using proactive behaviors taken by a failed robot that continue to support the completion of the task for which the human operator is responsible *task support*.

A robot operator's situation awareness (SA) is their ability to perceive information related to the state of the system and its surroundings, comprehend that knowledge within the robot's current context, and project or anticipate future events [Endsley, 1995]. As autonomy increases, the time the system can run while being ignored by the operator (known as neglect-time) also increases [Olsen and Wood, 2004]. This decreases the amount of attention the operator pays to the system at any given moment until ultimately the system is considered unsupervised. When a problem occurs in such a system, the person or people responsible for the robot's operation find themselves lacking sufficient SA to either understand the current problem or identify the appropriate actions that need to be taken - a phenomenon known as the out-of-the-loop problem [Kaber and Endsley, 1997]. When an autonomous system has been designed to provide information to people that supports or improves their SA with respect to the failure and the status of the task being performed, we say the system is providing *human support*.

3.1 Experiment

A previous investigation of failure mitigation strategies looked at using recovery strategies from the context of consumer research to improve users' satisfaction after a robot fails [Lee et al., 2010]. This included giving users advanced warning that

the robot might fail due to the difficulty of a task, having the robot apologize, offering compensation (such as a refund), and offering alternative options. Their success with these techniques may be related to attribution theory - that consumers try to infer the cause of a failure, and their conclusions drive their expectations for how a situation should be handled [Folkes, 1984]. This can lead to further dissatisfaction if the way a situation is handled does not match the consumer's expectations [Andreassen, 2000], and suggests that satisfaction with how a situation is handled can be controlled by ensuring that people have good situation awareness about the cause of failure.

Hypothesis 3.1: *Providing human support will help mitigate the negative effects caused by failure.*

When using a fully autonomous robot, an operator entrusts a task or responsibility to the system that they expect to be carried out. The relationship between the operator and the system can be thought of as a form of delegation since many tasks require the use of some level of discretion while being carried out. Thus, behaviors that work towards the completion of the task should be viewed favorably, especially if the robot is otherwise unable to complete the task itself.

Hypothesis 3.2: *Providing task support will help mitigate the negative effects caused by failure.*

Human and task support could have unintended consequences. Human support implemented using speech could result in unrealistic expectations that the robot is also capable of some form of task support [Cha et al., 2015]. Moreover, performing task support without providing sufficient human support could cause confusion. Combining the two techniques should minimize these kinds of problems without negative side effects.

Hypothesis 3.3: *A combination of both human and task support will help*

mitigate the negative effects caused by failure.

As the negative effects of a failure are reduced, positive sentiments towards the robotic service should increase.

Hypothesis 3.4: *Recovery strategies which reduce the negative effects of a failure will also increase the likelihood of users wanting to use the system again.*

3.1.1 Survey Design

We conducted two between-subjects survey studies, approved by the Institutional Review Board at the University of Massachusetts Lowell, to test our hypotheses. Our studies were modeled on the technique used in [Lee et al., 2010]. Participants were presented with a short two part story about a fictional character "Chris" who in one survey used a vacuum cleaner robot and in the other a self-driving taxi. The first part gave a brief background of Chris and included a short history of Chris' previous experience with the robot (reported in a positive manner). The second part described Chris' most recent encounter with the robot and the results of that interaction.

3.1.2 Independent Variables

Four independent variables were manipulated in this study: *context risk*, *failure severity*, *task support*, and *human support*. *Context risk* (risk) referred to how undesirable a failure by the robot would be in a particular context or setting, and was either "high" or "low." *Failure severity* (severity) referred to the type of failure the robot experienced and the extent to which it would be an inconvenience. It was either "none" (no failure occurs), "low," or "high." The robot either had *task support* and/or *human support* capabilities, or it did not. Combinations that did not involve failure but included *task support* were not tested, as we were unable to

Table 3.1: Independent variable combinations of survey conditions

Context Risk (`r`)	Severity (`s`)	TaskSupport (`t`)	HumanSupport (`h`)
low	low	yes	yes
high	low	yes	yes
low	high	yes	yes
high	high	yes	yes
low	none	no	yes
high	none	no	yes
low	low	no	yes
high	low	no	yes
low	high	no	yes
high	high	no	yes
low	low	yes	no
high	low	yes	no
low	high	yes	no
high	high	yes	no
low	none	no	no
high	none	no	no
low	low	no	no
high	low	no	no
low	high	no	no
high	high	no	no

conceptualize any scenarios in which this combination made sense. These variables were combined into twenty survey conditions for each robot scenario, as shown in Table 3.1.

The variables were represented in the story text in different forms to make the scenarios realistic. In the vacuum scenario, Chris was simply experimenting with new settings on the robot to expand the area it would clean for "low" context risk. For "high" risk, Chris was portrayed as a "neat-freak" relying on the robot to clean the house before having guests arrive, despite having never previously attempted this. When the failure severity was "None," the vacuum worked as Chris intended it to. "Low" failure severity was manifested by the robot not having enough battery to complete the job and Chris returning home to find the floors only partially cleaned. Finally, "High" failure severity depicted the robot creating an additional mess by knocking over a house plant. In scenarios where the robot did not have enough battery to complete cleaning, task support allowed the robot to return to

its charger and later (some time after Chris had returned home) resume cleaning from where it left off. The robot without task support would simply clean as long as possible until it ran out of batteries and died in the middle of the floor. In scenarios where the robot knocked over the house plant, the robot with task support would continue cleaning but avoid the area immediately around the accident so as not to make matters worse. In contrast, the robot without task support would attempt to drive through the area resulting in further damage to the plant (tearing off leaves) and spreading mud around the carpet. Human support was implemented by allowing the robot to send status updates about its progress to Chris. The method by which the robot communicated was intentionally omitted and left to the reader's imagination, with the exception of the robot being depicted as able to remotely notify Chris at work.

For the taxi scenario, Chris was going to the grocery store for "low" risk and to the airport to catch a flight for "high" risk. When the failure severity was "none," the vehicle worked exactly as Chris anticipated. During the "low" severity condition, the vehicle attempts to pass a slowly moving vehicle ahead of it while on the highway and misses the exit it was supposed to take. In the "high" severity condition, severe weather interrupts the vehicle's ability to drive and it pulls over on the side of the road. Task support during "low" severity conditions has the vehicle reroute along the next fastest available route to the destination. Without task support the vehicle reroutes itself to turn around and go back to location it originally got off route at, despite a faster route being available. In the "high" severity condition, the vehicle with task support automatically calls for a human-driven vehicle to come to the location the vehicle is stopped to take the passenger to their destination. Without task support, Chris has to summon a new ride. When the vehicle has human support, a map with route information and the vehicle's location is dis-

played, an estimated arrival time is shown and updates are provided (after the failure occurs), and information about recovery actions being taken are reported. Additionally, in the "high" severity condition human support provides a warning message stating that the vehicle is unable to operate in severe weather, and (if not combined with task support) informs the passengers that they need to find another ride.

3.1.3 Dependent Variables

We measured 9 dependent variables using a series of 7 point Likert scale questions regarding how participants believed the character (Chris) felt about the robot following the second half of the story. Participants were asked how *satisfied, pleased*, and *disappointed* Chris was with the service. They were asked how *reliable, dependable, competent, responsible*, and *trustworthy* Chris believed the robot to be. Finally, they were asked how *risky* it would be for Chris to use the robot in the future (see Table 3.2). Anticipating that any kind of failure might overpower the effects of the other independent variables, participants were asked to compare Chris' latest experience relative to previous experience with the robot using the scale *Much Less, Less, Somewhat Less, About the Same, Somewhat More, More,* and *Much More* for each dependent variable. Each variable was measured twice using two differently worded questions. The wording of the questions was kept consistent between scenarios, with the exception of context relevant words.

Participants were also asked two questions related to how they personally felt about the robot. These included whether they would want to use the robot described in the story, and if they would recommend the robot in the story to a friend.

Table 3.2: Vacuum Scenario Questions

How much more or less . . .
. . . satisfied is Chris with the robot's performance now compared to previous experiences?
. . . pleased is Chris with the robot's most recent results compared to previous experiences?
. . . does Chris trust the robot now compared to prior use?
. . . trust does Chris now have in the robot, compared to previous experiences?
. . . will Chris rely on the robot to clean the floors in the future?
. . . dependable does Chris believe the robot to be compared to before?
. . . competent does Chris believe the robot to be compared to before?
. . . certain is Chris that the robot will be able to clean the whole house in the future, given this latest experience?
. . . responsible does Chris believe the robot to be compared to before?

Possible Responses: *Much Less, Less, Somewhat Less, About the Same, Somewhat More, More,* and *Much More*

3.1.4 Manipulation and Attention Checks

Four "attention check" questions where included to check that participants were paying careful attention to the survey. After reading each of the two parts of the story, participants were asked a multiple choice question the answer to which would be obvious to anyone that had read the story - such as "What was the name of the character in the story?" In addition, two attention check questions were included in the bank of Likert questions to ensure people were carefully reading the questions. The answers to these questions were included in the question itself, such as "How much more or less does Chris take pictures? Please answer 'less' to this question." Failure to answer any of the attention check questions resulted in disqualification of the data for analysis.

Participants were also asked to answer six true or false style questions about things mentioned during the story, called manipulation check questions. The questions asked about details in the story related to the four independent variables. Participant's needed to answer all six of these questions correctly in order to demonstrate they had correctly perceived the various important aspects of the story, and have their data included for analysis.

Table 3.3: Demographics

Age

	Vacuum	Taxi
18-21	15	13
22-34	247	281
35-44	150	148
45-54	103	98
55-64	65	49
65-over	20	11

Gender

	Vacuum	Taxi
Male	294	289
Female	304	306
Other	2	5

Education

	Vacuum	Taxi
< HS	4	3
HS	70	49
Vocational	19	26
In College	148	130
2 Yr Deg	74	63
4 Yr Deg	211	238
Grad Deg	74	91

3.2 Results

Data was gathered using Amazon's Mechanical Turk, with each participant being paid $0.90 for their work. Participants consisted of self-selected MTurk workers who lived in the United States and had previously performed at least 1000 Human Intelligence Tasks (HITs) with at least a 95% approval rating. We collected 30 participants worth of complete data for each condition in each scenario, totaling 600 participants for each type of robot and a combined total of 1200 participants. We were able to facilitate a between-subjects study due to MTurk workers being required to register their tax information with their account, MTurk providing unique workers for each HIT, and disallowing individual IP addresses from completing each scenario more then once. While 68 of the 1200 people involved (5.6%) participated in both the taxi and vacuum scenarios, each scenario was analyzed independently. See Table 3.3 for demographic details.

3.2.1 Measuring reaction to failure

Each of the dependent variables reflected different aspects of participants' overall perception of the character's (Chris') reaction to the robot's latest performance. We performed an exploratory factor analysis of the Likert scale questions for each scenario. A Scree test concluded that in both cases there was a single latent

Table 3.4: Factor Analysis Variable Loadings

Variable	Vacuum	Taxi
p.satisfied	0.94	0.83
p.n.satisfied	0.68	0.58
p.pleased	0.94	0.90
p.n.pleased	0.67	0.59
p.trust1	0.94	0.92
p.trust2	0.96	0.92
p.reliable1	0.94	0.92
p.reliable2	0.94	0.90
p.dependable1	0.96	0.90
p.dependable2	0.95	0.92
p.competent1	0.94	0.86
p.competent2	0.91	0.87
p.responsible	0.91	0.78
p.n.responsible	0.78	0.55
p.disappointed1	-0.78	-0.68
p.disappointed2	-0.77	-0.69
p.risky1	-0.84	-0.75
p.risky2	-0.83	-0.85

variable. The factor analysis accounted for 77% of the variance in the vacuum data and 66% of the variance in the taxi data. Variables in the taxi scenario had a Chronbach's $\alpha = 0.97$ and variables in the vacuum scenario had a Chronbach's $\alpha = 0.98$. All variables loaded the single factor, which we call `REACTION`, in both cases (see Table 3.4). Responses to questions with negative wordings were inverted prior to analysis; however, the negative attributes "disappointed" and "risky" were not inverted and subsequently received negative loadings. Thus, positive scores represent positive reactions to the robot's behavior while negative scores represent negative reactions. `REACTION` scores are shown in Figure 3.1.

A two-way ANOVA was performed to compare the influence of failure severity and context risk on `REACTION`. There was a significant main effect of failure severity on `REACTION` in both the taxi $[F(2) = 287.1284, p < 0.001, \eta_p^2 = 0.491]$ and vacuum $[F(2) = 410.4056, p < 0.001, \eta_p^2 = 0.58]$ surveys. A significant main effect of context risk on `REACTION` was found in the taxi survey $[F(1) = 13.6936, p < 0.001, \eta_p^2 = 0.039]$, but not in the vacuum survey $[F(1) = 0.9546, p = 0.33, \eta_p^2 = 0.0007]$.

There was a significant interaction between context risk and failure severity in taxi survey[$F(2) = 7.6941, p < 0.001, \eta_p^2 = 0.025$], but not in the vacuum survey [$F(2) = 0.8814, p = 0.415, \eta_p^2 = 0.0029$].

Most participants who experienced the robot failing without any support had a negative `REACTION` (*taxi:* 92%, n=120; *vacuum:* 90%, n=120), while nearly everyone who experienced the robot without any failure (both with and without support) had a positive `REACTION` (*taxi:* 99%, n=120; *vacuum:* 96%, n=120).

3.2.2 Effect of support on reaction

A one-way ANOVA was performed to compare the influence of support type on people's reactions for each risk-failure combination. There was a significant effect of support type on `REACTION` in all conditions in both robot scenarios at a significance level of $\alpha = 0.05$. In the taxi scenario, a significant main effect was found in the low-risk, low-failure condition [$F(3, 116) = 12.61, p < 0.001, \eta^2 = 0.246$], the low-risk, high-failure condition [$F(3, 116) = 15.27, p < 0.001, \eta^2 = 0.283$], the high-risk, low-failure condition [$F(3, 116) = 3.805, p < 0.05, \eta^2 = 0.089$], and the high-risk, high-failure condition [$F(3, 116) = 12.19, p < 0.001, \eta^2 = 0.239$]. In the vacuum scenario, a significant main effect was found in the low-risk, low-failure condition [$F(3, 116) = 13.16, p < 0.001, \eta^2 = 0.254$], the low-risk, high-failure condition [$F(3, 116) = 17.27, p < 0.001, \eta^2 = 0.309$], the high-risk, low-failure condition [$F(3, 116) = 38.06, p < 0.001, \eta^2 = 0.496$], and the high-risk, high-failure condition [$F(3, 116) = 21.42, p < 0.001, \eta^2 = 0.356$]. For each of these conditions, a post-hoc Tukey's HSD test was used to determine significant differences between human support (`HUMAN`), task support (`TASK`), combined human and task support (`COMBINED`) and no support (`NONE`). The results of the post-hoc tests are shown in Figure 3.1.

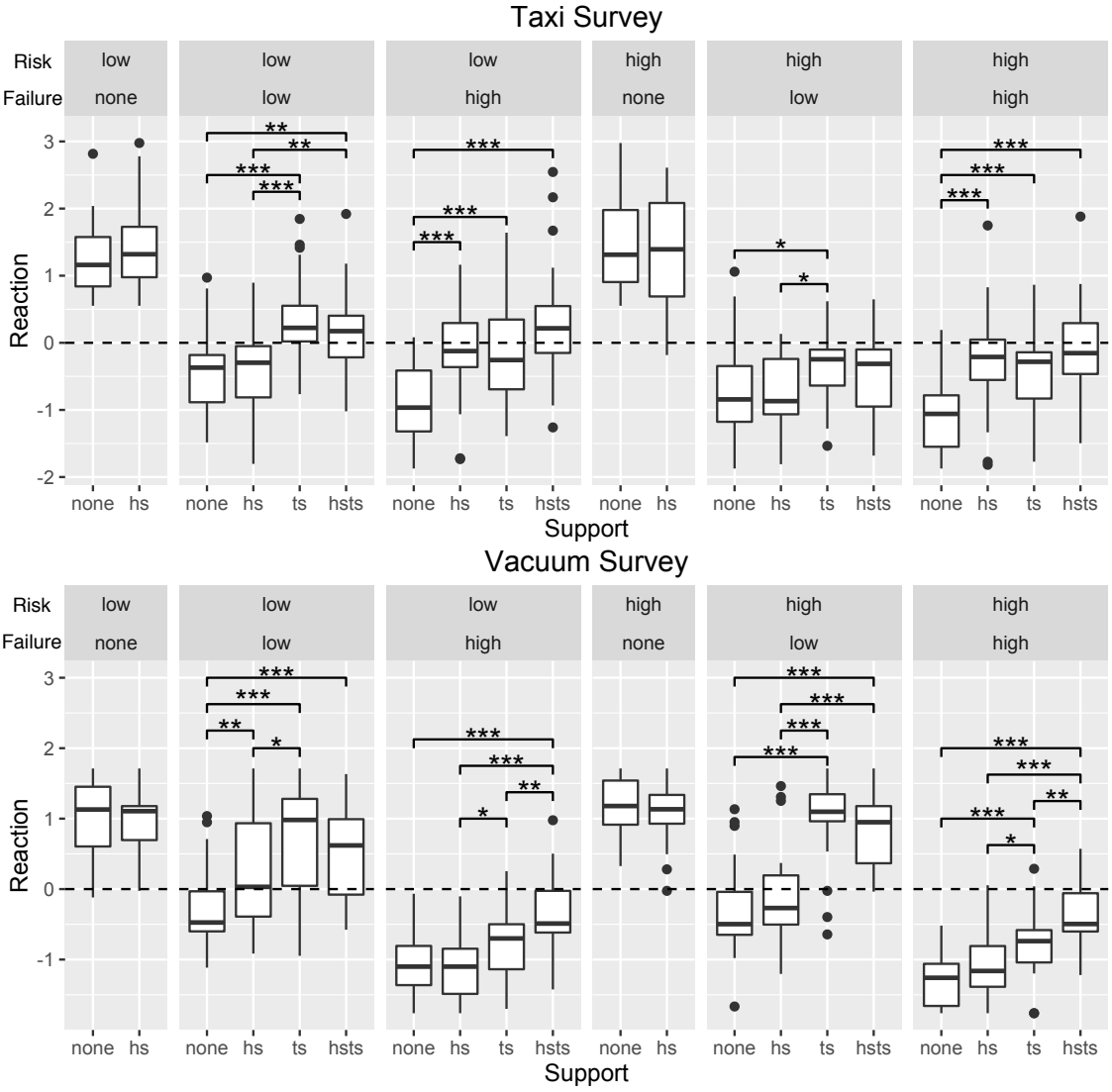


Figure 3.1: Participants' `REACTION` scores grouped by risk, failure, and support type. *hs:* `HUMAN`, *ts:* `TASK`, *hsts:* `COMBINED`. $n = 30$ for each bar. Results from post-hoc Tukey's HSD tests: $\star p \leq 0.05$, $\star\star p < 0.01$, $\star\star\star p < 0.001$.

3.2.3 Effect of support on wanting to use the robot

A one-way ANOVA was performed to compare the influences of support on people's responses to wanting to use the robot described in the scenario they read. There was a significant main effect of support on wanting to use the robot in all conditions. In the vacuum scenario, there was a significant main effect [$F(3, 116) = 4.53, p < 0.01, \eta^2 = 0.105$] in the low-risk, low-failure condition. A post-hoc test showed significant differences between `TASK` and `NONE` ($p < 0.05$), and `COMBINED` and `NONE` ($p < 0.01$). There was a significant main effect [$F(3, 116) = 4.25, p < 0.01, \eta^2 = 0.099$] in the high-risk, low-failure condition. A post-hoc test showed significant differences between `TASK` and `NONE` ($p < 0.05$), and `COMBINED` and `NONE` ($p < 0.05$). There was a significant main effect [$F(3, 116) = 3.876, p = 0.01, \eta^2 = 0.091$] in the low-risk, high-failure condition. A post-hoc test showed significant differences between `TASK` and `HUMAN` ($p = 0.05$), and `COMBINED` and `HUMAN` ($p = 0.01$). There was a significant main effect [$F(3, 116) = 4.905, p < 0.01, \eta^2 = 0.112$] in the high-risk, high-failure condition. A post-hoc test showed significant differences between `COMBINED` and `NONE` ($p < 0.01$), and `COMBINED` and `HUMAN` ($p < 0.05$).

In the taxi scenario, there was a significant main effect [$F(3, 116) = 3.339, p = 0.02, \eta^2 = 0.079$] in the low-risk, low-failure condition. A post-hoc test showed significant differences between `TASK` and `NONE` ($p = 0.02$). There was a significant main effect [$F(3, 116) = 4.695, p < 0.01, \eta^2 = 0.108$] in the high-risk, low-failure condition. A post-hoc test showed significant differences between `TASK` and `NONE` ($p < 0.05$), `COMBINED` and `NONE` ($p < 0.05$), `TASK` and `HUMAN` ($p < 0.05$), and `COMBINED` and `HUMAN` ($p < 0.05$). There was a significant main effect [$F(3, 116) = 5.139, p < 0.01, \eta^2 = 0.117$] in the low-risk, high-failure condition. A post-hoc test showed significant differences between `HUMAN` and `NONE` ($p < 0.01$), and `COMBINED` and `NONE` ($p = 0.01$). There was a significant main effect [$F(3, 116) = 7.016, p <$

$0.001, \eta^2 = 0.153$] in the high-risk, high-failure condition. A post-hoc test showed significant differences between HUMAN and NONE ($p < 0.01$), COMBINED and NONE ($p < 0.001$), and COMBINED and TASK ($p < 0.05$).

The REACTION score of each participant was compared to their response for "I would want to use this robot/vehicle." 95% (251/264) of participants in the vacuum survey and 77% (200/258) of participants in the taxi survey who had a positive REACTION score responded with some level of agreement. Of participants who had a negative REACTION, only 58% (194/336) of participants in the vacuum survey and 42% (144/342) of participants in the taxi survey responded with some level of agreement.

3.3 Analysis and Findings

Participants' REACTION was significantly influenced by failure severity in both the taxi and vacuum surveys, and by context risk in the taxi survey. The REACTION scale correctly divided people who experienced the robot operating successfully from people who experienced the robot failing (with no support) by whether or not their score was positive or negative with 94% accuracy ($n = 480$). The magnitude of people's REACTION was significantly influenced by the severity level of the failure in both surveys.

The REACTION scale also highlights the variability by which recovery strategies can alter a person's response to a failure, ranging from having no measurable effect to being indistinguishable from not having failed. Further, it indicates that the effectiveness of recovery strategies (which in general improved people's REACTION to failure) seems to be influenced by the task, context risk, and severity or type of failure.

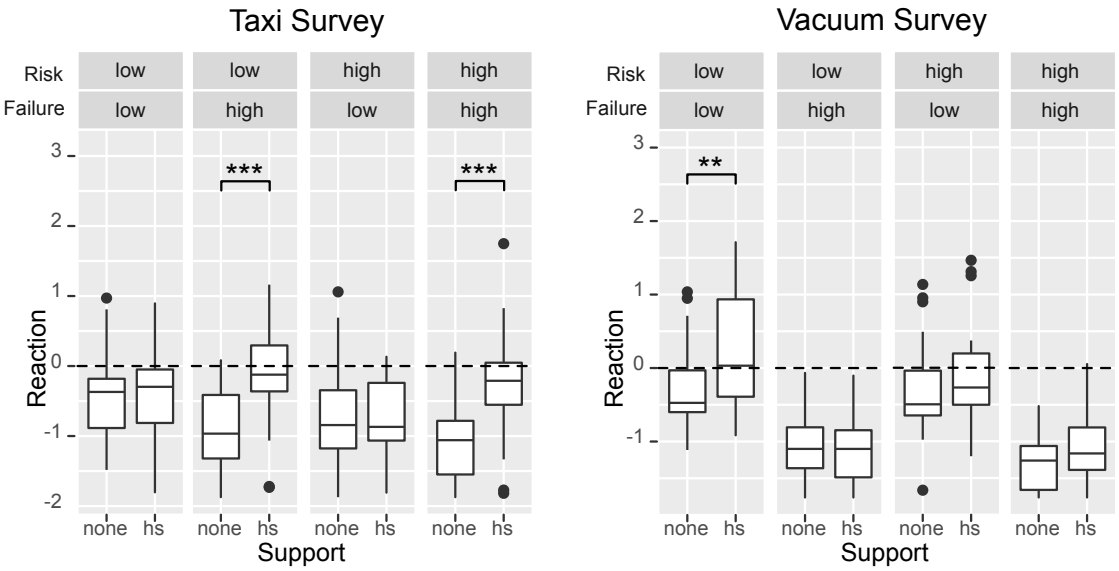


Figure 3.2: Effectiveness of Human Support compared to no support

Hypothesis 3.1: *Providing human support will help mitigate the negative effects caused by failure.*

This hypothesis was partially supported by our results. Human support significantly ($p < 0.01$) improved people's `REACTION` in several scenarios (see Figure 3.2). The amount it influenced people's `REACTION` varied by the task, severity of failure, and context risk. However, the significance of human support seems to be better correlated to whether the information conveyed could be used by the person to affect the outcome of the situation. In the high severity condition of the taxi scenario, the car informed the passenger they needed to call for another ride, which significantly improved people's `REACTION`. When the taxi missed a turn in the low severity condition, the support information allowed the user to predict but not affect the outcome, and had almost no effect. There were no conditions in the vacuum scenario in which the human support was used to alter the outcome of the situation. However, it still significantly improved people's reactions in the low-risk, low-failure scenario. This could be interpreted as Chris knowing that he

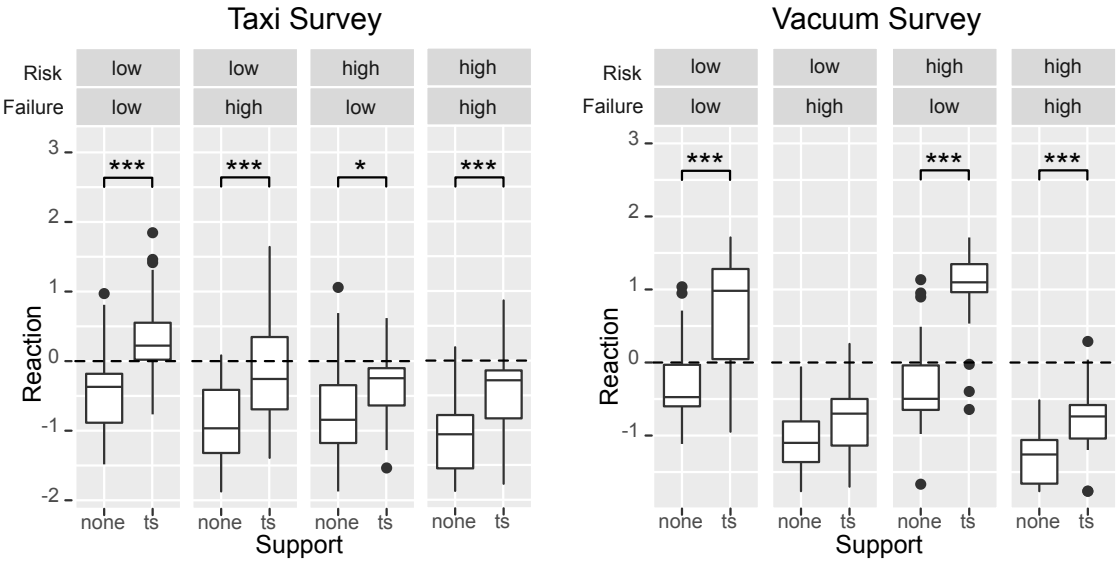


Figure 3.3: Effectiveness of Task Support compared to no support

would need to clean the floor himself when he got home - something he would have the chance to do in the low risk scenario but not the high risk scenario.

Hypothesis 3.2: *Providing task support will help mitigate the negative effects caused by failure.*

This hypothesis was supported by our results. Using task support significantly improved people's `REACTION` ($p < 0.05$) in all but one scenario (vacuum, low-risk, high-failure severity, $p = 0.06$). One particularly interesting data point is the extremely positive `REACTION` to task support in the high-risk, low-failure condition of the vacuum scenario. The robot's behavior in this case was to return to its charger before the battery ran out, and resume cleaning where it left off when it had recharged - thus eventually completing the task. While the completed task certainly contributed to the high `REACTION`, the response to the same behavior in the corresponding low-risk condition had a much higher variance. One possible explanation is that the difference in variance may be the result of people being less certain about the significance of the failure in the low-risk condition compared to

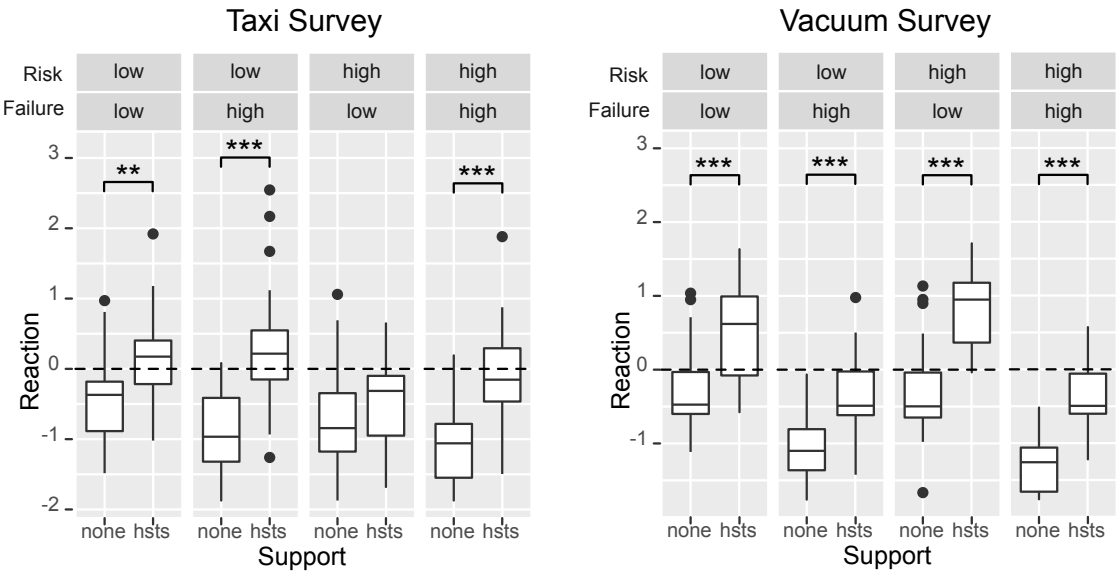


Figure 3.4: Effectiveness of Combined Human and Task Support compared to no support

the high-risk condition. However, this would suggest there should also be higher variances in the low-risk, high-failure condition, which was not the case. Another possible explanation is that the difference in risk changed the way people imagined Chris perceiving the way the task was completed. In the low-risk condition Chris was portrayed as experimenting with the robot's capabilities, which may have prompted a more critical view of the results, while in the high-risk condition Chris was hoping for a particular result despite the lack of a precedent, making the robot's performance a pleasant surprise.

Hypothesis 3.3: *A combination of both human and task support will help mitigate the negative effects caused by failure.*

This hypothesis was largely supported by our results. Combined support significantly ($p < 0.001$) improved people's `REACTION` in all but one scenario (taxi, high-risk, low-failure severity, $p = 0.16$). In one case (vacuum, high-risk, high-failure), combined support was significantly better ($p < 0.01$) than using task support, which was itself significantly better ($p < 0.001$) than no support. How-

ever, a non-significant trend can be seen in which combined support performed better than task support in high severity situations, but worse in low severity situations. One possible explanation for this is that in certain situations the additional information is regarded as too verbose or possibly annoying, while in others the information is welcomed. Unfortunately, this logic would be better supported if the trend corresponded to differences in context risk rather than the failure severity.

Hypothesis 3.4: *Recovery strategies which reduce the negative effects of a failure will also increase the likelihood of users wanting to use the system again.*

This hypothesis was supported by our results. The percentage of people who wanted to use the robot was much higher among people with a positive `REACTION` score than among those who had negative scores. Both human and task support affected how much people wanted to use the robot, although neither effect was ubiquitous.

3.4 Limitations to this study

Survey experiments have some inherent flaws [Visser, Krosnick, and Lavrakas, 2000]. The self-selection of participants may have introduced non-response error into the data, and the extensive use of rating scales in our survey may have caused some people to mark multiple questions with the same answer (non-differentiation). Responses could be biased for various reasons such as acquiescence response bias (tendency to agree regardless of the question) or question wording incidentally cueing a particular response. Our survey was only available to people residing in the US, and may not reflect the way people in other parts of the world would behave in similar situations. Furthermore, prior work has shown the MTurk population does not perfectly match the US population (it was also not extremely different)

[Berinsky, Huber, and Lenz, 2012]. Thus, the experiment could benefit from being repeated with other populations.

Finally, the third person perspective of both the story and questions was chosen over a first person perspective to allow participants to distance themselves from the situation, reducing the effects of subconsciously biased responses such as from people trying to portray themselves in a particular manner [Nisbett, Caputo, Legant, and Marecek, 1973]. However, reading about a hypothetical situation someone else is experiencing is not the same as experiencing the same situation for one's self in real life. Thus, a laboratory experiment in which participants experience failures in-person is needed to verify these results.

3.5 Conclusions

The `REACTION` scale captures the main characteristics of failure by autonomous robots, while also highlighting the nuanced complexity of the situation. We have demonstrated its use by comparing successful and failed operation of robots with various recovery strategies. In doing so we found evidence that while human support and task support can both be used to mitigate failures, the type and severity of failure, and context risk influence their effectiveness.

In this study, we only compared results of the `REACTION` scale within individual robots due to the use of separate exploratory factor analyses for each study. Similarities observed between factor loadings of the two analyses suggest we should be able to refine the `REACTION` scale into a generic question bank that will be task- and platform- independent, potentially allowing it to be used as a comparison between robots.

Chapter 4

Identifying Platform-Independent Robot Status Icons

A person's situation awareness (SA) is their ability to perceive the world around them, comprehend that knowledge in the context of the current situation, and predict or project future events [Endsley, 1995]. When a person is unable to determine what a robot is currently doing or is uncertain about what it will do next, we describe them as having poor SA. Lack of SA can lead to serious safety concerns; it can also impact people's perception of the system and their trust of it. As a robot's autonomy increases, the length of time it can run while being ignored by an operator (neglect time) also increases [Olsen and Goodrich, 2003; Olsen and Wood, 2004]. This leads to the out-of-the-loop problem [Kaber and Endsley, 1997], in which the people who would normally be the most informed about the system (the operator or user) find themselves lacking sufficient SA to either understand the current problem or identify the appropriate actions to be taken. Bystanders, who are unassociated with the robot other than by virtue of being co-present with it [Scholtz, 2003; Yanco and Drury, 2004], can be expected to have even worse SA

Figure 4.1: Robot drawings from design surveys. (left to right) *Self-driving car, floor cleaning robot, package delivery quadcopter, generic mobile helper robot.*

than the robot's operator or user due to being unfamiliar with the system.

In this chapter, we investigate the plausibility of an interaction style by which bystanders might gain SA of autonomous robots through the use of icons that could be standardized for use across all robots. Such icons would be displayed externally on the body of a robot as a method of conveying simplified information about an autonomous robot's internal system state. We focus on attempting to convey information about two particular categories of information which both a robot's operators/users and bystanders would identify as being important: *Is the robot safe to be around?* and *Is the robot working properly?* These two categories were then broadened into a series of five abstract pieces of information that we refer to as our target messages.

OK: The robot is fine / ok / operating properly.

HELP: The robot needs help / assistance.

OFF: The robot is turned off / not in operation.

SAFE: The robot is safe to be around / safe to approach / won't hurt you.

DANGEROUS: Stay away from the robot / keep back / do not approach / dangerous.

As the first step, our goal was to determine if a small set of icons exist that can convey the same information across different robots as a proof of concept. We performed a series of online surveys to identify potential candidate icons and

experimentally tested their ability to convey information to untrained participants. Our results indicate that icons are a viable method for communicating information from a wide variety of robot platforms to untrained observers.

4.1 Robot State Icons

Our objective was to determine if a single set of icons could represent our target messages on many different kinds of robots. Our hypotheses were as follows:

Hypothesis 4.1: *Icons can be used to intentionally communicate state information of an autonomous robot to a bystander.*

Hypothesis 4.2: *People will share a single, predictable, interpretation of each icon.*

Hypothesis 4.3: *The characteristics of the robot an icon is presented on will not alter the interpreted meaning of the icon.*

Lacking a set of clear guidelines for the creation of icons to represent our target messages, we first conducted a series of three online surveys (S1-S3) to crowdsource a test set (Section 4.2). These crowdsourced icons were then tested in an experiment where we displayed them on various robots to see how their presence affected people's interpretations of those machines (Section 4.3). All work was conducted using Amazon's Mechanical Turk. The icon design and experiment surveys were approved by our Institutional Review Board.

4.2 Crowdsourcing a Set of Icons

The icons used for our experiment in Section 4.3 were iteratively designed across three surveys. In each survey, participants were shown a line drawing of one of four

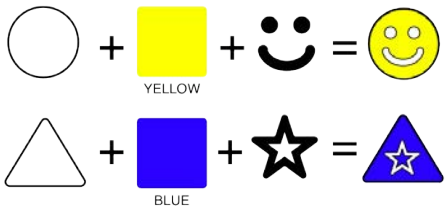


Figure 4.2: Icon components example shown in S1

types of autonomous robots (Figure 4.1) along with a brief paragraph explaining the purpose of the pictured robot and that in the near future encountering robots such as the one shown would become a more common occurrence in daily life. They were then told that these robots will need to communicate basic information to people around them, and the participant's job was to help us identify icons that could be illuminated on the sides of the robot to convey particular meanings.

Each survey contained a number of "attention check" questions to provide a minimal level of quality control. The answers to these questions were not subjective; if answered incorrectly, the participant was immediately disqualified from the survey and information collected from them was discarded during analysis as being unreliable.

4.2.1 Icon Attributes - Survey 1

Survey 1's (S1) objective was to identify and rank which colors, symbols, and shapes people associated with our target messages. Participants were told that we planned to construct icons by combining shapes, colors, and symbols and shown the example in Figure 4.2 as illustration. First, participants were asked a series of 5 questions in which they were instructed to *select the five best symbols which you think would mean ...* followed by one of the target messages. Figure 4.3 shows the set of symbols participants were asked to choose from. In addition to these five questions, there was a check question which asked participants to *select the two*

Figure 4.3: Symbols (*left*) and shapes (*right*) from S1

symbols that have been drawn in the shape of a person's hand.

Next, participants were shown a series of questions in which they were asked to rearrange a set of objects being displayed to place them in order from "best" to "worst" to indicate how well each represented a particular message. The objects were displayed as a vertical list that could be reordered by dragging objects up or down. Questions were divided into five pages, one for each target message, each with three questions for symbols, shapes, and colors respectively. The ordering of the pages was randomized, as were the order of the questions on each page and the initial ordering of the objects in each question. The questions involving symbols displayed the symbols previously selected by the participant for that message, while questions involving shapes presented the participant with a set of 6 shapes to choose from (see Figure 4.3) and questions involving colors involved the set of blue, green, red, and yellow. Finally, a second attention check question was added, asking participants to *please drag the shapes to arrange them in order of the number of sides on each shape, assuming the speech bubble has seven sides, and the circle has one. Place the lowest number at the top, and the highest number at the bottom.* The square shape was removed from the check question to eliminate the ambiguity of ordering two four sided objects.

4.2.1.1 Survey 1 Results

We collected data from over 500 unique participants in this survey. Of those, 65 voluntarily withdrew and 32 were rejected for failing the attention check. It took

Figure 4.4: Icon candidates for the OK message, generated from Survey 1 and used for Survey 2.

most people less then 10 minutes to complete the survey, and those who finished were paid $0.90. We performed our analysis using data collected from the first 120 participants in each of the four robot examples[1], giving us a total of 480 unique participants in this survey. Participants consisted of 174 women, 231 men, 1 who reported as "other", and 74 who chose not to report. Age groups consisted of 14 people from 18-21, 209 from 22-24, 98 from 35-44, 51 from 45-54, 27 from 55-64, 7 over 65, and 74 did not report. Forty one had a high school degree, 96 had some college (no degree), 164 had bachelors degrees, 47 had graduate degrees, 42 had associates degrees, 15 had vocational training, and 75 chose not to respond.

Scores were computed for each color, shape, and symbol with respect to each target message using an inverse weighted sum based on the number of times the object was ranked by participants in each position. The color red was primarily associated with the DANGEROUS and OFF messages, green with SAFE and OK, and yellow with HELP (Figure 4.5). For shapes, octagons were most highly associated with DANGEROUS and OFF, with triangles representing a distant second in both cases. Circles were highly associated with SAFE and OK, with speech bubbles being a distant second in both cases. However, two shapes (triangle and speech bubble), were both closely associated with HELP (Figure 4.5).

[1]The number of desired responses was set for each condition in advance. Our software sometimes collected too many responses; additional responses were excluded from analysis.

4.2.2 Combined Components - Survey 2

Using the information we collected in S1, we assembled 5 groups of icons, each of which we believe could potentially represent a target message. For example, our information indicated that participants favored the color green, circles and speech bubbles, and the symbols of a check mark, happy face, play arrow, star, and thumbs up for the `OK` message. S1 did not allow for solid shapes that lacked an internal symbol in the previous survey. Therefore, we also included "blank" icons which were based only on the shapes and colors used to generate the other icons. Figure 4.4 shows the icons generated for the target message `OK` as an example.

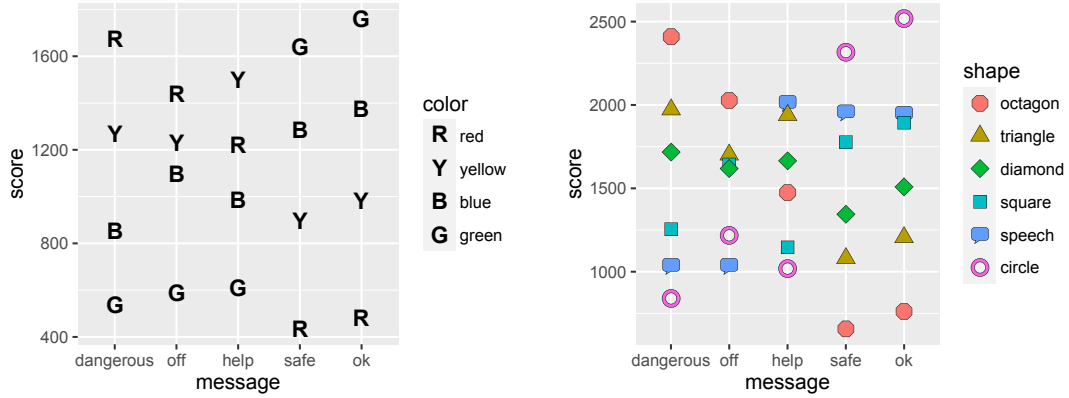


Figure 4.5: Color Results (*left*) and Shape Results (*right*) from Icon Survey 1

By asking people to consider individual components of an icon, we ignored the possibility that an icon created by combining those components may not represent the same message as the individual parts by themselves. For example, someone may think that both the color green and a power button symbol represent that a robot is turned on, but associate a green colored power button with being used to turn on a robot that is currently powered off. The goal of S2 was to identify how well our generated icons represented the target messages and identify the icons with the most potential.

Participants were asked to *select the five best icons which you think would mean...* for each of the five target messages. For each message, the icon choices were based on the data collected from S1 for that message. The ordering of the questions and choices were randomized. Additionally, a check question was added that asked participants to *select the six icons from the set below that are in the shape of a triangle* from the set of icons created for the `HELP` message.

Similar to the second half of S1, participants were next asked to rank their previous answers by rearranging a vertical list of icons and placing them in order from "best" to "worst" according to how well each represented the message. There were a total of six randomly ordered questions, one for each target message plus a check question. Each question had the five previously selected icons for that message, initially arranged in a random order. We also included a check question that showed a unique set of five different shaped icons, and asked participants to *drag the icons to arrange them in the order of circle, triangle, diamond, speech bubble, octagon. Place the circle at the top, and the octagon at the bottom.*

4.2.2.1 Survey 2 Results

Over 250 people participated in our second survey. Of those, 12 voluntarily withdrew before completing the survey and 37 were rejected for failing the attention check. Most people took less then 10 minutes; those who finished were paid $0.90. We performed our analysis on data from the first 50 unique participants in each of the four robot examples, for a total of 200 unique participants. Participants consisted of 83 women and 117 men. Age groups consisted of 7 people from 18-21, 114 from 22-34, 45 from 35-44, 20 from 45-54, and 14 from 55-64. Twenty one had a high school degree, 55 had some college (no degree), 72 had bachelors degrees, 23 had graduate degrees, 20 had associates degrees, 7 had vocational training, and

Figure 4.6: Icons tested during Icon Survey 3

2 chose not to report.

Icons were evaluated based on how well they represented each target message individually. Scores for each icon were calculated using an inverse weighted sum based on the number of times participants ranked that icon in each of the top five positions for each message. Icon scores for individual messages were then compared against each other, with icons positioned in the upper quartile for each message being selected for use in S3. A total of 17 icons were selected (3 *ok*, 3 *safe*, 3 *help*, 4 *dangerous*, 4 *off*); however, due to a few icons being selected multiple times the final set consisted of 15 unique icons (see Figure 4.6).

4.2.3 Validation of Icons - Survey 3

Survey 3 (S3) was used to determine if the icons selected from S2 (Figure 4.6) could be used to accurately convey their intended messages to people, and whether there were significant differences in how well various icons represented each target message. Instead of showing participants messages and asking them to select the best icon as in S2, we now showed them an icon and asked what they thought it meant.

During this survey, participants were presented with each of the icons, one at a time in random order, and asked to select which message(s) each was meant to convey from a multiple choice list that included the five target messages (randomly arranged) plus a sixth option allowing people to write in a message that wasn't on the list. Each question was worded as *"Suppose you see the following icon being*

Icon	OK	SAFE	HELP	OFF	DNGRS
circle.green.check	287	294	120	122	120
circle.green.thumb	267	344	120	120	120
circle.green.happy	243	368	120	120	120
speech.green.happy	224	368	129	120	120
speech.yellow.exclaim	123	153	319	126	149
speech.yellow.question	122	147	352	120	120
triangle.yellow.exclaim	120	153	304	133	160
triangle.red.skull	120	152	127	123	345
triangle.red.power	120	147	133	331	130
triangle.red.hand	120	148	143	129	330
octagon.red.x	120	150	151	217	234
octagon.red.skull	120	152	130	121	342
octagon.red.minus	120	151	145	280	171
octagon.red.hand	120	148	130	132	336
octagon.red.power	120	150	127	321	144

Table 4.1: Icon Survey 3 Sum of Ranks

displayed on the exterior of a robot. Which of the following messages is the robot trying to convey to you? You may select one or more options." Included in these questions was an attention check question in which participants were shown a blue circle with a star in it (the only blue icon or icon with a star being shown) and asked to select from a list of icon descriptions the one that best matched the icon displayed; the answer described the icon itself.

4.2.3.1 Survey 3 Results

We collected data from over 150 participants. Of those, 13 voluntarily withdrew and 10 were rejected for failing the attention check. The survey took about 15 minutes, and those who finished were paid $1.50. We analyzed data collected from the first 30 unique participants in each robot example, totaling 120 unique participants. Participants consisted of 48 women, 71 men, and 1 person who choose not to report. Age groups consisted of 3 people from 18-21, 69 from 22-34, 27 from 35-44, 15 from 45-54, and 6 from 55-64. Nine had a high school degree, 36 had some college (no degree), 45 had bachelors degrees, 11 had graduate degrees, 18 had associates degrees, and 1 had vocational training.

Responses for each icon were evaluated with respect to individual messages. For each message, the multiple choice questions were recoded as ranked values according to how well responses matched the target message. For example, the target message `HELP` was coded as a 3 if "the robot needs help" was the only message the participant believed the robot was trying to indicate with the icon. If the participant also selected one or more additional meanings, it was coded as 2. If one or more meanings were selected but did not include "the robot needs help" it was coded as 1, and if no meanings were selected it was coded as 0.

To determine if certain icons were better than others at conveying certain information, a Friedman's analysis of variance by ranks was performed for each message on the resulting scores of the 15 icons. Differences across conditions were significant for `OK` [$\chi^2(14) = 1359.1, p < 0.001$], `SAFE` [$\chi^2(14) = 1215.1, p < 0.001$], `HELP` [$\chi^2(14) = 1196.9, p < 0.001$], `OFF` [$\chi^2(14) = 1210.2, p < 0.001$], and `DANGEROUS` [$\chi^2(14) = 1264.7, p < 0.001$]. The sum of ranks for each condition are shown in Table 4.1.

Green icons were best received for *ok* and *safe* while red was associated with *off* and *dangerous*. The circular shape tended to be most associated with good messages while triangles and octagons were associated with problems.

4.3 Experiment

A set of five icons which performed particularly well during S3, one for each target message, was selected for use in an experiment to test our hypotheses (listed in Section 4.1). We selected *circle.green.check* for `OK`, *speech.yellow.question* for `HELP`, *triangle.red.power* for `OFF`, *circle.green.happy* for `SAFE`, and *octagon.red.skull* for `DANGEROUS` (Figure 4.7).

Figure 4.7: Final Icons: `OK`, `HELP`, `OFF`, `SAFE`, `DANGEROUS`

During the experiment, photos[2] of robots were manipulated to make it appear as though our icons were displayed on LCD displays on the robots. These photos were shown to participants, who were then asked several questions concerning what they thought about the robot shown to them.

4.3.1 Methodology

We manipulated two independent variables, the icon being displayed and the robot it was displayed on. Five icons plus a condition in which no icon was displayed ("`DEFAULT`") were tested on 14 different robot platforms representing various robot sizes, appearances, capabilities, methods of locomotion, and form factors (Table 4.2). Several different types of robot platforms which were not represented during the design process were included to reduce the likelihood of overfitting and validate the generalizability of our test set. Additionly, we tested the effects of displaying state icons on robots that have a face by creating two different images of the Baxter platform, one that had Baxter's default eyes on the screen and one that had the manufacturer's logo. This gave us a total of 6 icon conditions and 90 different images.

Images were selected to meet our requirements: the robot had to appear to be motionless, not have any people that could be seen, have as few lights or indicators as possible, be near an object for scale, and have a flat surface pointed towards the camera for the icon to be placed on. The icon size in the image had to be between

[2]Taken by the authors or publicly available and licensed.

7.5% and 8.5% of the total image width to ensure icon visibility was consistent across images. Only image width was used to determine icon size since images scaled to the window or screen size.

Our dependent variables were people's perception of whether the robot was enabled or disabled, was safe or dangerous to be around, "was working", or needed help/assistance. After being shown an image, participants were presented with the following four 7-pt Likert scale questions (in random order).

- The robot depicted in the picture is dangerous to be around.
- The robot depicted in the picture has been disabled.
- The robot depicted in the picture is operating properly.
- The robot depicted in the picture needs assistance.

Responses to these questions were *Strongly Disagree, Disagree, Somewhat Disagree, Neither Disagree nor Agree, Somewhat Agree, Agree, Strongly Agree*.

The experiment was advertised as a study about what people perceive in images and consisted of three tasks on separate pages, presented in random order, so as not to portray the study as being specifically about robots. In the *robot task*, participants were shown a single image of a robot and asked about their agreement on a series of questions. This was the only task from which we collected data. In the *airplane task*, participants were shown a randomized grid of 15 images depicting modes of transportation and asked to select all images that showed airplanes. This task was used as an attention check question (similar to a captcha). In the *nature task*, participants were shown a single image, such as a sunset or a mountain, and asked to select from a list of 18 adjectives which ones they believed applied to the picture. This task was a distraction task and not analyzed.

4.3.2 Experiment Results

More then 2900 people participated. Of those, 97 voluntarily withdrew and 78 were rejected for failing the attention check. The survey took about 2 minutes, and those who finished were paid $0.20. Analysis was performed using data collected from the first 30 participants for each of the 90 images, totaling 2700 unique participants (1388 women, 1299 men, 7 reported as other, and 6 who did not report). Age groups consisted of 75 people from 18-21, 1420 from 22-34, 643 from 35-44, 340 from 45-54, 179 from 55-64, and 43 over 65. Two hundred and sixty four participants had a high school degree, 585 had some college (no degree), 1086 had bachelors degrees, 357 had graduate degrees, 300 had associates degrees, 88 had vocational training, and 20 chose not to respond. Few people reported having any experience with the robot they were shown (except for Roomba).

4.3.2.1 Overall effect of the icons

Participant responses for all 15 robots were combined to perform 4 one-way ANOVA to compare the influences of icons (`OK`, `SAFE`, `HELP`, `OFF`, `DANGEROUS`, and `DEFAULT`) on participants' perception of the danger of being around the robot, if the robot had been disabled, if it was operating properly, and if it needed assistance (see Figure 4.9).

Danger: There was a significant main effect on the perceived danger [$F(5, 2694) = 16.72, p < 0.001, \eta_p^2 = 0.03$]. A post-hoc test found significant differences between `DEFAULT`—`DANGEROUS` ($p < 0.001$) and `DANGEROUS`—`OK`,`SAFE`,`HELP`,`OFF` ($p < 0.001$). Additional significant differences were found between `OFF`—`SAFE` ($p = 0.002$) and `OFF`—`OK` ($p < 0.03$).

Disabled: There was a significant main effect on the robot being perceived as disabled [$F(5, 2694) = 32.65, p < 0.001, \eta_p^2 = 0.057$]. A post-hoc test found

significant differences between DEFAULT—OFF ($p = 0.03$), and OFF—OK,SAFE,HELP ($p < 0.001$). Significant differences were also found between DEFAULT—OK, SAFE ($p < 0.001$), and OK,SAFE—HELP,DANGEROUS ($p < 0.001$).

Operating Properly: There was a significant main effect on the robot being perceived as operating properly [$F(5, 2694) = 32.82, p < 0.001, \eta_p^2 = 0.057$]. A post-hoc test found significant differences between DEFAULT—OK,SAFE ($p < 0.001$) and OK,SAFE—OFF,HELP, DANGEROUS ($p < 0.001$). Additionally, a significant difference was found between DEFAULT—DANGEROUS ($p = 0.016$).

Needs Assistance: There was a significant main effect on the robot being perceived as needing assistance [$F(5, 2694) = 20.71, p < 0.001, \eta_p^2 = 0.037$]. A post-hoc test found significant differences between DEFAULT—HELP ($p = 0.043$) and HELP—OK,SAFE ($p < 0.001$). Additionally, significant differences were found between DEFAULT—OK,SAFE ($p < 0.01$), DEFAULT—OFF ($p = 0.016$), and OFF—OK,SAFE ($p < 0.001$). Finally, significant differences were found between DEFAULT—DANGEROUS ($p = 0.046$) and DANGEROUS—OK,SAFE ($p < 0.001$).

4.3.2.2 Robot Characteristics

Twelve robots (marked in Table 4.2) were selected prior to data collection as a subset which balanced the number in each condition for each characteristic. The Quad Drone and DARPA Car were not included due to their unique nature (flying and being a human transport) and not being able to provide enough additional examples to keep the conditions balanced. Participant responses from these 12 robots were used to perform 4 separate factorial ANOVA to compare the influences of the icons, size (small, medium, or large), appearance (industrial or consumer), manipulators (yes or no), mobility (rolling, legged, or ambiguous), and form factor (machine or biologically inspired) on participants' perception of the robot being

dangerous, disabled, operating properly, and needing assistance. Only individual factors and first-order interactions involving icons are discussed. Tukey's HSD test was used to compute p-values.

Dangerous: There was a significant main effect of the icon shown [$F(5) = 17.823, p < 0.001, \eta_p^2 = 0.04$], size [$F(2) = 76.802, p = 0.035, \eta_p^2 = 0.04$], appearance [$F(1) = 156.983, p < 0.001, \eta_p^2 = 0.02$], manipulators [$F(1) = 20.493, p < 0.001, \eta_p^2 = 0.004$], and mobility [$F(2) = 8.854, p < 0.001, \eta_p^2 = 0.008$], but not form factor [$F(1) = 0.046, p = 0.829, \eta_p^2 < 0.001$], on if the robot was perceived as being dangerous. No significant interactions were reported between the icon shown and size [$F(10) = 1.773, p = 0.06, \eta_p^2 = 0.006$], appearance [$F(5) = 0.992, p = 0.421, \eta_p^2 = 0.002$], manipulators [$F(5) = 1.481, p = 0.19, \eta_p^2 = 0.003$], mobility [$F(10) = 0.761, p = 0.667, \eta_p^2 = 0.004$] or form factors [$F(5) = 0.359, p = 0.87, \eta_p^2 < 0.001$].

Disabled: There was a significant main effect of the icon shown [$F(5) = 42.023, p < 0.001, \eta_p^2 = 0.09$], size [$F(2) = 3.347, p = 0.035, \eta_p^2 = 0.013$], manipulators [$F(1) = 80.718, p < 0.001, \eta_p^2 = 0.069$], mobility [$F(2) = 117.661, p < 0.001, \eta_p^2 = 0.1$], and form factor [$F(1) = 11.450, p < 0.001, \eta_p^2 = 0.005$], but not appearance [$F(1) = 0.344, p = 0.55, \eta_p^2 = 0.019$], on if the robot was perceived as being disabled. There were significant interactions reported between the icon shown and size [$F(10) = 2.327, p = 0.01, \eta_p^2 = 0.003$], appearance [$F(5) = 7.001, p < 0.001, \eta_p^2 = 0.013$] and mobility [$F(10) = 2.544, p < 0.005, \eta_p^2 = 0.01$], but not between the icon shown and manipulators [$F(5) = 0.653, p = 0.66, \eta_p^2 < 0.001$] or form factor [$F(5) = 1.755, p = 0.12, \eta_p^2 = 0.004$].

Operating Properly: There was a significant main effect of the icon shown [$F(5) = 36.514, p < 0.001, \eta_p^2 = 0.079$], size [$F(2) = 21.504, p < 0.001, \eta_p^2 = 0.001$], manipulators [$F(1) = 45.93, p < 0.001, \eta_p^2 = 0.043$], and mobility [$F(2) =$

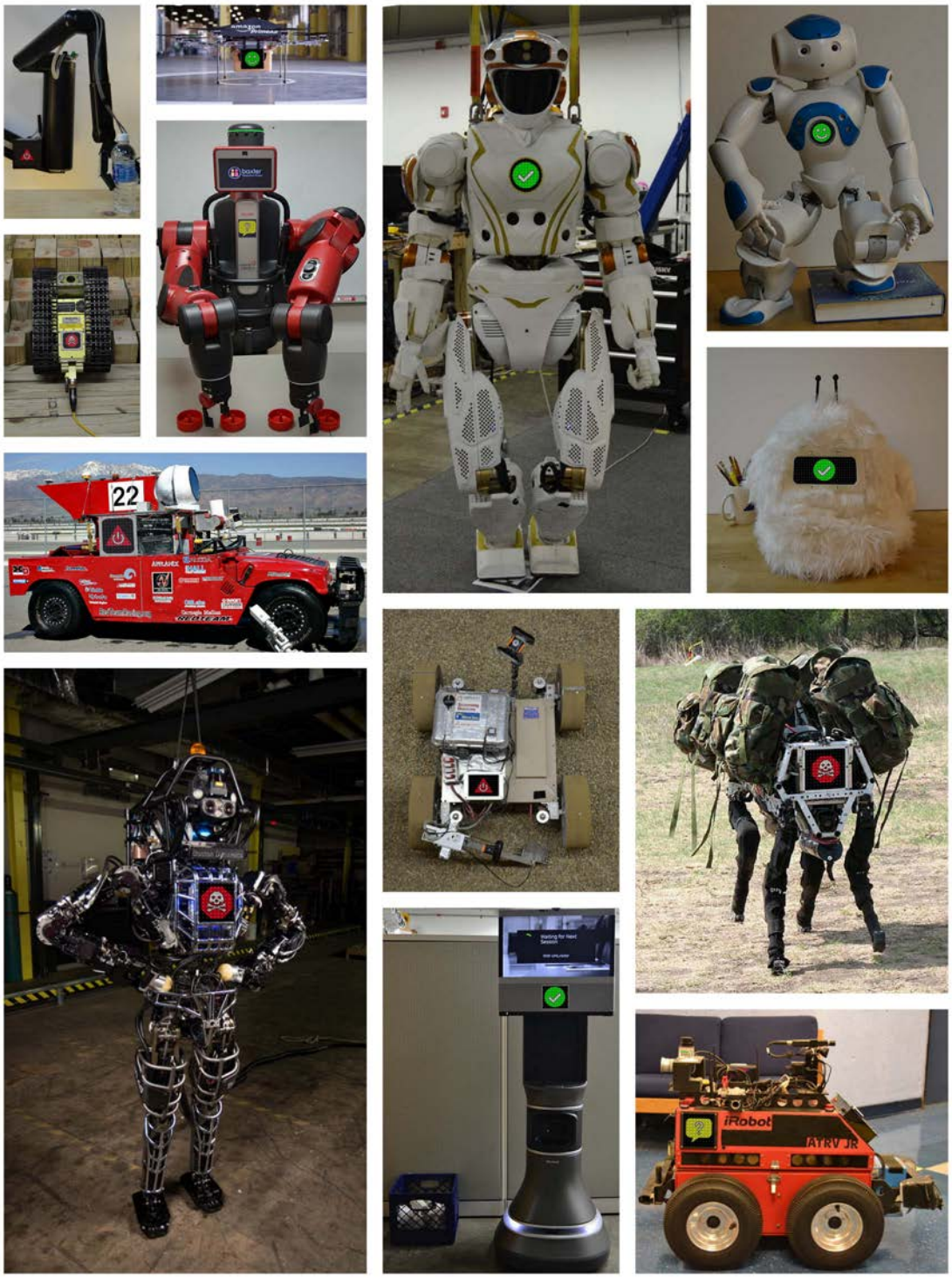


Figure 4.8: A few of the photoshopped images used during the experiment.

$107.37, p < 0.001, \eta_p^2 = 0.093]$, but not appearance $[F(1) = 0.86, p = 0.35, \eta_p^2 = 0.007]$ or form factor $[F(1) = 2.594, p = 0.11, \eta_p^2 = 0.001]$, on if the robot was perceived to be operating properly. There were significant interactions reported between the icon and size $[F(10) = 2.657, p = 0.003, \eta_p^2 = 0.003]$, appearance $[F(5) = 5.31, p < 0.001, \eta_p^2 = 0.012]$, and form factor $[F(5) = 2.56, p = 0.025, \eta_p^2 = 0.006]$, but not between the icon and manipulators $[F(5) = 0.901, p = 0.48, \eta_p^2 = 0.001]$ or mobility $[F(10) = 1.641, p = 0.093, \eta_p^2 = 0.007]$.

Needs Assistance: There was a significant main effect of the icon $[F(5) = 23.417, p < 0.001, \eta_p^2 = 0.05]$, size $[F(2) = 7.193, p < 0.001, \eta_p^2 = 0.009]$, manipulators $[F(1) = 22.745, p < 0.001, \eta_p^2 = 0.03]$, and mobility $[F(2) = 89.74, p < 0.001, \eta_p^2 = 0.079]$, but not appearance $[F(1) = 3.451, p = 0.63, \eta_p^2 = 0.003]$ or form factor $[F(1) = 3.231, p = 0.072, \eta_p^2 = 0.001]$, on if the robot was perceived as needing assistance. There were significant interactions reported between the icon and size $[F(10) = 4.157, p < 0.001, \eta_p^2 = 0.003]$ and appearance $[F(5) = 6.718, p < 0.001, \eta_p^2 = 0.016]$, but not between the icon shown and manipulators $[F(5) = 0.586, p = 0.71, \eta_p^2 < 0.001]$, mobility $[F(10) = 1.054, p = 0.39, \eta_p^2 = 0.005]$, or form factor $[F(5) = 1.119, p = 0.35, \eta_p^2 = 0.002]$.

4.3.2.3 Comparison of Baxter with and without a Face

We investigated whether the presence of a human-like face on a robot effects how people perceive status icons being displayed. Two versions of the Baxter robot were tested, one in which a set of eyes were displayed on the monitor and one in which the Rethink Robotics logo was displayed. We performed 4 separate two-way ANOVA to compare the influence of the presence of eyes and state icon on participant's perception of whether the robot was dangerous, disabled, operating properly, or needed assistance.

Dangerous: There was no significant main effect of the presence of the eyes $[F(1) = 0.879, p = 0.349]$ or the icon shown $[F(5) = 1.238, p = 0.291]$ on whether the robot was perceived to be dangerous, nor was there a significant interaction between the presence of the eyes and the icon shown $[F(5) = 0.696, p = 0.627]$.

Disabled: There was no significant main effect of the presence of the eyes $[F(1) = 0.026, p = 0.872]$ or the icon shown $[F(5) = 1.761, p = 0.120]$ on whether the robot was perceived to be disabled, nor was there a significant interaction between the presence of the eyes and the icon shown $[F(5) = 1.688, p = 0.137]$. A Two-tailed Welch's T-Test found a significant difference between the robot with eyes and the one without eyes when the `SAFE` icon was shown $[t(56.889) = 2.001, p = 0.05]$, however the Tukey HSD post-hoc analysis indicated the difference was not significant $(p = 0.66)$.

Operating Properly: There was significant main effect of the icon shown $[F(5) = 2.389, p = 0.037]$ on whether the robot was perceived to be operating properly, but not for the presence of the eyes $[F(1) = 2.754, p = 0.098]$, nor was there a significant interaction between the presence of the eyes and the icon shown $[F(5) = 1.484, p = 0.194]$. A Tukey HSD post-hoc analysis showed a significant difference only between the `OK` and `DANGEROUS` icons $(p = 0.04)$. A Two-tailed Welch's T-Test found a significant difference between the robot with eyes and the one without eyes when the `SAFE` icon was shown $[t(53.152) = 2.96, p = 0.004]$, however the Tukey HSD post-hoc analysis indicated the difference was not significant $(p = 0.144)$.

Needs Assistance: There was no significant main effect of the presence of the eyes $[F(1) = 0.320, p = 0.572]$ or the icon shown $[F(5) = 0.597, p = 0.703]$ on whether the robot was perceived to need assistance, nor was there a significant interaction between the presence of the eyes and the icon shown $[F(5) = 0.338, p = 0.890]$.

Platform	Size	Appearance	Arm	Mobility	Form
Roomba*	Small	Consumer	No	Rolling	Mach.
ATRV Jr*	Med	Industrial	No	Rolling	Mach.
Baxter*	Large	Consumer	Yes	Ambig.	Bio
Valkyrie*	Large	Consumer	Yes	Legged	Bio
Nao*	Small	Consumer	Yes	Legged	Bio
Atlas*	Large	Industrial	Yes	Legged	Bio
Manus Arm*	Med	Industrial	Yes	Ambig.	Mach.
iRobot Ava*	Med	Consumer	No	Ambig.	Mach.
Romibo*	Small	Consumer	No	Ambig.	Bio
Rover Hawk*	Med	Industrial	Yes	Rolling	Mach.
VGTV*	Small	Industrial	No	Rolling	Mach.
BigDog*	Large	Industrial	No	Legged	Bio
Quad Drone	Small	Consumer	No	Flying	Mach.
DARPA Car	Large	Industrial	No	Rolling	Vhcle

Table 4.2: Robot platforms shown in Experiment. Those with * were used to evaluate characteristics.

4.3.3 Discussion

Our results supported H4.1: icons can convey state information to bystanders. People who were shown `DEFAULT` images mostly believed that the robots were not dangerous. This result was similar to the group who experienced the `SAFE` icons, while those who experienced the `DANGEROUS` icons were much less certain. People shown the `OK` and `SAFE` icons were more likely to believe the robot was operating properly and less likely to believe it was disabled compared to those shown `DEFAULT`, `HELP`, `OFF`, or `DANGEROUS`. Those who were shown `OFF` were more likely to believe it was disabled compared to the `DEFAULT` images, while people shown the `HELP` icon were more likely to believe the robot needed assistance.

Our results partially supported H4.2 and H4.3: people would share a single, predictable interpretation of each icon and that robots' characteristics would not alter the interpreted meaning of the icons. People interpreted multiple meanings from single icons, possibly from extrapolating the consequences of one meaning to another (e.g. if the robot is disabled, it will also need assistance in the form of being turned back on). People in the `HELP`, `OFF`, and `DANGEROUS` conditions were all more likely to believe the robot need help than those in `DEFAULT`, `OK`, and `SAFE`

conditions. Unlike DANGEROUS, those in HELP believed the robot was safe and were less likely than those in OFF to believe the robot had been disabled.

However, the expected meanings of icons were both predictable and additional interpretations never conflicted with the expected meaning. Participants in the OFF and DANGEROUS conditions were much more likely to believe the robot had been disabled and less likely to believe it was operating properly than those OK condition. However, those in OFF were also less likely than participants in DANGEROUS to believe the robot was dangerous to be around. The strength of the responses for icons varied based on robot characteristics, but never resulted in a significant change between meanings.

4.3.3.1 Dangerous

In the DEFAULT images, people were significantly more likely to agree the robot was dangerous if it was large compared to if it was medium sized ($p = 0.03$) or small ($p < 0.001$). This makes sense; it is reasonable to associate large (and presumably heavy) machinery as posing a greater threat than smaller machines. Robots with an industrial appearance were also perceived as more dangerous than those with a consumer appearance ($p < 0.001$). However, the mobility, presence or absence of manipulators, and form factor did not seem to have an effect on perceived danger.

There was also a significant difference between people shown the DANGEROUS icon and the DEFAULT image for small robots ($p < 0.001$), regardless of appearance ($p \leq 0.001$), the method of mobility ($p \leq 0.05$), presence of manipulators ($p \leq 0.003$), or the form factor ($p \leq 0.001$). People shown the DEFAULT images of smaller robots (such as the Roomba) strongly believed they were not dangerous, while people who were shown the same image but with the DANGEROUS icon were much more likely to believe that the robot was dangerous.

People's perception of if the robot was dangerous was significantly higher when they were shown the DANGEROUS icon compared to when they were shown the OK or SAFE icons for large and small robots ($p \leq 0.002$). Notably absent from these results are medium sized robots, which is difficult to explain. People shown the DANGEROUS icon were also significantly more likely to agree that the robot was dangerous than those who were shown the OK, SAFE, or HELP icons regardless of the robot's appearance ($p \leq 0.02$), method of mobility ($p < 0.02$, except for *legged* robots), presence of manipulators ($p \leq 0.007$), or form factor ($p \leq 0.001$). Additionally, the DANGEROUS icon was significantly different from the OFF icon when manipulators were present ($p = 0.008$) or the robot had a machine-like form factor ($p = 0.041$).

4.3.3.2 Disabled

In the DEFAULT images, people were much more likely to perceive the robot as disabled if it had manipulators ($p = 0.004$), or if the method of mobility was rolling compared to legged ($p = 0.005$) or ambiguous ($p < 0.001$). The robot's size, appearance, and form factor did not significantly affect the perception of the robot being disabled in DEFAULT images.

Robots with the OK icon were less likely to be viewed as disabled than those in the DEFAULT images regardless of size ($p < 0.01$ for large and small, but not medium), appearance ($p < 0.01$), method of mobility ($p < 0.001$ for rolling and legged, but not ambiguous), the presence of manipulators ($p < 0.001$), or form factor ($p < 0.001$). The SAFE icon had the same effect but was more muted, with significant differences only seen with small robots, those with a consumer appearance, and those that rolled, regardless of manipulators or form factor ($p < 0.001$).

Participants shown images of the robot with the OFF icon were more likely to believe the robot was disabled than those shown the OK or SAFE icons regardless

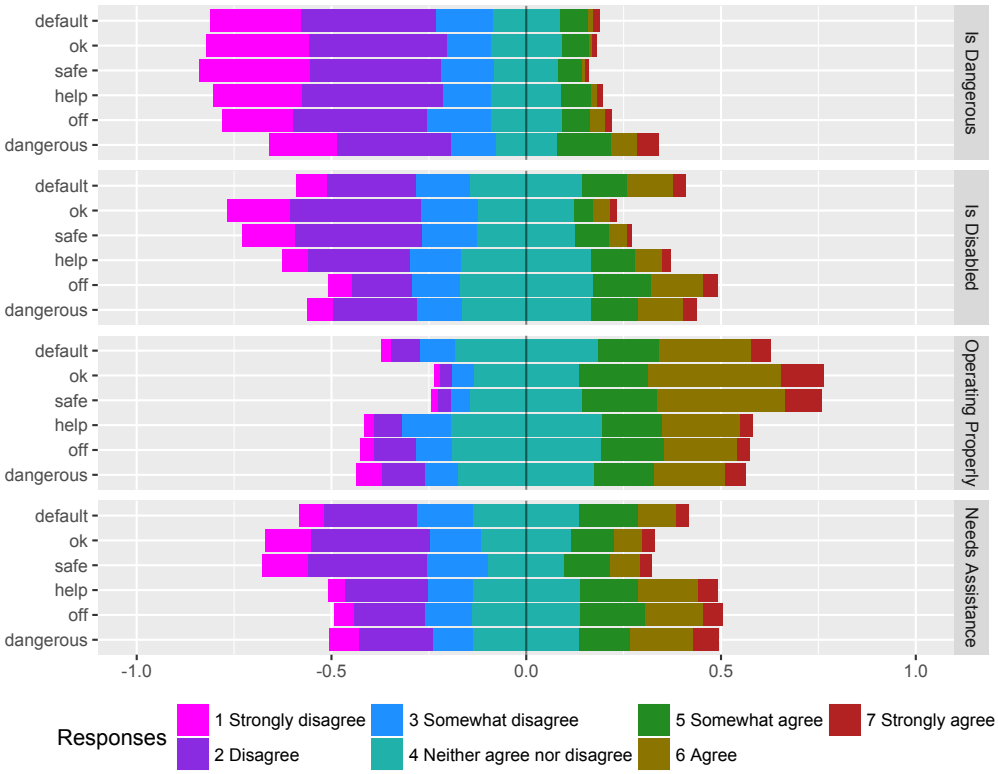


Figure 4.9: Experiment Results

of size ($p < 0.003$), appearance ($p < 0.001$), presence of manipulators ($p < 0.001$), form factor ($p < 0.001$), or method of mobility ($p < 0.001$, except `OFF` vs. `SAFE` for ambiguous mobility). The `HELP` icon was seen as less likely to be disabled than `OFF` across all characteristics, with significant differences found for small robots ($p = 0.002$), robots that rolled ($p = 0.005$), or had manipulators ($p < 0.001$), regardless of appearance ($p < 0.04$) or form factor ($p < 0.02$). The same effect was observed much more weakly between `HELP` and the `OK` and `SAFE` icons . Robots with the `HELP` icon were seen as more likely to be disabled than those with `OK` across all characteristics, with significant differences found for small robots ($p < 0.001$), robots with a consumer appearance ($p < 0.001$), robots with legs ($p = 0.02$), both manipulator conditions ($p <= 0.01$), and those with a machine-like form factor ($p < 0.001$). The `HELP` icon was also seen as more likely to be disabled than those with the `SAFE` icon across all characteristics, with significant differences found for

small robots ($p = 0.016$), robots with a consumer appearance ($p < 0.001$), robots that rolled ($p = 0.023$), those without manipulators ($p = 0.002$), and those with a machine-like form factor ($p = 0.002$). Finally, people were generally more likely to assume that robots with the DANGEROUS icon were disabled compared to the OK or SAFE icons ($p < 0.01$ in all conditions across all characteristics, except for SAFE with large or industrial looking robots).

4.3.3.3 Operating Properly

In the DEFAULT images, people were much more likely to perceive large robots as operating properly than small robots ($p = 0.02$). This was also true for rolling robots compared to legged ($p = 0.005$) or ambiguous ($p < 0.001$) methods of mobility. Differences in appearance, manipulators, or form factor were not significant in the DEFAULT images.

Robots with the OK and SAFE icons were more likely to be viewed as working than those in the DEFAULT images of small robots, those with a consumer appearance, and robot's that moved by rolling, regardless of the presence of manipulators or the form factor ($p \leq 0.006$ in all comparisons).

Participants shown images of robots with the OK or SAFE icons were more likely to believe the robot was operating properly than people who were shown the OFF or DANGEROUS icons regardless of size ($p \leq 0.054$), appearance ($p < 0.02$), method of mobility ($p \leq 0.02$), presence of manipulators ($p < 0.001$), or form factor ($p < 0.001$). The OK and SAFE icons also generated more positive responses regarding the robot working properly compared to the HELP icon for large and small robots (but not medium sized) ($p \leq 0.011$) and robots with rolling or legged mobility methods ($p \leq 0.004$), regardless of an appearance ($p < 0.05$), presence of manipulators ($p < 0.001$), or form factor ($p \leq 0.001$).

4.3.3.4 Needs Assistance

In the DEFAULT images, people were much more likely to perceive rolling robots as requiring assistance than those with legs or ambiguous mobility methods ($p \leq 0.02$). The robot's size, appearance, presence of manipulators, and form factor did not have a significant effect on the robot being perceived as needing assistance in the DEFAULT images.

Images of small robots and those with a consumer appearance that showed the HELP icon were more likely to be viewed as needing assistance than those in the DEFAULT images ($p = 0.05$). This may be due to people perceiving small and well "polished" robots to be more relatable and less imposing. Alternatively, these results might be indicative of how people perceive themselves as being able to help robots; a smaller robot which could be easily lifted would be more likely to ask them for help than a large robot that would be difficult to move. At least one documented case already exists where a robot's size influenced people's interaction around it [Rae, Takayama, and Mutlu, 2013]. Robots with DANGEROUS icons were also much more likely to be viewed as needing assistance than their DEFAULT counterparts in the cases of small robots ($p = 0.013$), robots with a consumer appearance ($p < 0.001$), and those with a machine-like form factor ($p = 0.032$). Robots with the OK and SAFE icons were less likely to be seen as needing assistance, but the effect was less well defined. The SAFE icon was only significantly different from DEFAULT for small robots ($p = 0.011$), robots with a consumer appearance ($p = 0.029$), and rolling robots ($p = 0.041$), while OK was only significantly different from DEFAULT for small robots ($p = 0.015$).

People shown images of robots with the HELP, DANGEROUS, and OFF icons were more likely to believe the robot needed assistance than those who were shown the OK and SAFE icons, with significant differences found for small robots ($p < 0.001$),

those with a consumer appearance ($p < 0.001$), and rolling or legged mobility ($p \leq 0.014$), regardless of manipulators ($p \leq 0.008$) or form factor ($p \leq 0.006$). Industrial-looking robots with the `OFF` icon were also much more likely to be viewed as needing assistance than those with `OK` or `SAFE` ($p < 0.02$). Finally, robots with ambiguous mobility showing the `OK` icon were viewed as much less likely to need assistance than those with the `OFF` or `DANGEROUS` icons ($p \leq 0.031$).

The naive interpretation of these results suggests that the `DANGEROUS` or `OFF` icons are just as well suited as `HELP` at communicating that a robot needs assistance. However, both `DANGEROUS` and `OFF` were viewed as also expressing additional meanings. Our measurements may also simply reflect beliefs such as that robots which are working properly should not be dangerous or that a robot which has been disabled will require a person to turn it back on. The lack of effectiveness of the `HELP` icon in the ambiguous mobility condition is not easily explained, and merits further investigation.

4.4 Conclusions

Our results support the idea that a single set of icons can be used across many different robots to convey information to bystanders. Standardization of icons should increase people's recognition and comprehension of the icons' meanings and provide a reliable method for bystanders to gain SA when encountering unfamiliar robots.

While our results seem promising, there are some limitations that must be considered. Participants were restricted from performing each survey multiple times; however, we were unable to prevent people from participating in multiple surveys given the software used. The majority of our participants only took part in 1 of the 4 surveys (three design surveys plus the experiment); however, 220 participants

took part in 2 surveys and 12 participated in 3 of the 4.

Our experiment used static images which limited potentially conflicting information an observer would have in the real world. Robots convey information though motions and behaviors (e.g. shaking or running into walls), sounds (e.g. grinding or scraping), and location or position (e.g. high-centered on a curb or caught in a door). Participants who were shown the `DEFAULT` Ava platform were more likely to believe it was enabled than those who experienced other robots. We believe this was due to Ava having a large screen which was on, indicating it was waiting for a user to call in.

There were strong similarities between the `OK` and `SAFE` messages. In the introduction of each of the three design surveys, participants were inadvertently told that the robot was "able to operate safely around people." This may have biased participants' responses by associating working robots with being safe to be around, resulting in the high correlation observed between `OK` and `SAFE`. Despite this, there were still noticeable differences between the two messages. For example, in S3 *circle.green.check* was the highest scoring icon for `OK` but came in fourth for `SAFE`, while *speech.green.happy* shared the highest sum of ranks score for `SAFE` with *circle.green.happy*, but came in fourth for `OK`.

Our research was conducted only with participants from the United States. Symbols, shapes, and colors can have different meanings in different cultures, and even change within a culture over time. It is likely that if our experiment was repeated with a population from a different culture that it would produce different results. Also, our crowdsourced icon set was not necessarily optimal. For practicality, all of our icons were based on a limited set of options presented during S1, which effectively constrained the selection process.

Chapter 5

A Smartphone-based Interface for Ubiquitous Robot Communication

We have designed and built a smartphone-based interaction system which aims to allow co-located people and robots to directly exchange basic levels of information, both as a form of social courtesy and as a mechanism for improving people's situation awareness. Rather than a primary method of interacting with or controlling a robot, this system was designed as a universal secondary interface that could be used during (non-life-threatening) emergency situations or for impromptu communication between untrained users and potentially unfamiliar devices. This is accomplished by allowing nearby robots and smartphones to detect each other's presence through the use of a well-known protocol transmitted via radio signal. The demonstration system described in this chapter makes use of bluetooth low energy as an example. However, in the future the same system could be implemented in parallel across multiple communication modes (e.g., Wi-Fi, cellular data) for increased robustness.

The system works by having robots continuously advertise their presence to

nearby devices, such as smartphones, which acknowledge the robots by initiating a handshake and transfer of information. Robots provide information that can be used by people to physically identify them in the world along with status information, while smartphones offer cursory information about their owners' preferences. Following this initial handshake, information can be passed between the two systems until one of them moves out of range. The system is designed to support two different styles of interaction: one in which a person queries information about or initiates communication with a nearby robot, and another in which a robot reaches out to initiate communication with a nearby person. We refer to the former as *pull-style* interactions and the latter *push-style* interactions.

Pull-style interactions begin when a person wants to get information about or communicate with a nearby robot (i.e. they want to "pull up" information about a robot). The first step of this process is for the person to select the robot they wish to communicate with, either by opening a special smartphone app and selecting the target robot from a list of nearby robots, or by tapping on a background notification generated by the app. Robots are represented by an icon depiction of the hardware platform along with a unique identifier (which appears both on the smartphone and is physically written on the body of the robot.) Each robot's icon is paired with a robot's state icon representing the robot's overall status (see Figure 5.1). Once a robot has been selected, the user is shown a screen dedicated exclusively to displaying information about that specific robot, called the robot's "Status Page".

Robot status pages display the robot identification and information about its state that were shown when selecting the robot, additional information about the robot such as what it is currently doing, and (optionally) interactive dialogs that could be used to exert limited amounts of control over the system (see Figure 5.2). This information is displayed as a set of linear message boxes representing the flow

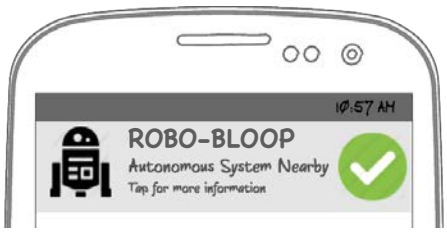


Figure 5.1: Example of a *pull-style* background notification. The notification is silently generated in the background, and can be accessed by swiping down the menu bar.

of a conversation (similar to a text message conversation) between the robot and the person, known as a "progression". Information provided by the robot can be appropriately balanced to provide surrounding people with some knowledge about what the robot is currently doing while also maintaining some level of privacy for the system's end user (when appropriate). People can communicate information back to the robot through a system of pre-specified options supplied by the robot. For example, in Figure 5.2, the robot gives people the option to relocate its position while it is waiting for a passenger. If this option is selected by a person, the response is sent back to the robot, which in turn sends that particular person a new dialog consisting of a map and asks the person to indicate a better location for the robot to move to.

Robots can specify whether information should be made available to everyone nearby (broadcast) or be directed to a specific individual. This feature is necessary, since interactions taking place between one individual and the robot may change the options that should be displayed to other people at the same time. For example, once the robot commits to allowing a person to relocate it, that option should be removed from other people until the new position has been established.

In certain circumstances, it may be appropriate for a robot to impose itself on a nearby person, interrupting them if necessary to get their attention (for example, to

Figure 5.2: Examples of a robot's status screen, showing the robot's ID, icon, state icon, and a more verbose status message including an option to relocate the robot *(left)*. If the person selects the option to move the robot, new dialog appears on the screen *(center)*. Other nearby robots can be selected from a menu accessible on a robot's status screen *(right)*.

warn of a dangerous situation or ask for help). We call this a *push-style* interaction (the information is being "pushed" on the person), as it is facilitated through disruptive notifications such as having the phone vibrate/make an audible sound and displaying a popup dialog on the screen. Popup dialogs contain the same icon of the robot and ID information displayed during *pull-style* interactions, a status icon representing the nature of the information, and possible responses that can optionally be sent back to the robot (see Figure 5.3). Information shown during *push-style* interactions is also made available on the robot's status page.

As an example of a *push-style* interaction, imagine a robot which needs to pass through a normally open doorway that has been closed (such as a fire door). If the robot finds a nearby group of people, it could use a *push-style* interaction to ask one person from the group if they would hold the door open for it. The request would include an iconified picture of the robot itself and its identification name so the device-user could associate the information as having come from the

Figure 5.3: A *Push-style* interaction. (*from left to right*) A *push-style* popup notification, responding "yes", response changed to "later", and a follow-up question to the answer "later".

sending robot. A help icon at the top would indicate that the robot is asking for assistance as opposed to warning them of some danger. Finally, a text description of the robot's request would be shown, along with an image of the particular door the robot would like opened. The person would be presented with dialog buttons allowing them to accept or decline the request, or even ask the robot to come back and ask them again in a few minutes. Meanwhile, the other people in the group would not have received the message from the robot since it only needed the help of one person. In a *pull-style* interaction, the robot's presence would still show up on the other people's smartphones in the list of nearby robots such that if any of them were to go looking for information about the robot and access its status page they would still see the same request for help. Each bystander would be able to access this information until the point that someone explicitly accepted the robot's request for help or the robot received the help it needed.

The rest of this chapter describes our development of a demonstration system we built by modifying commercially available autonomous robot vacuum cleaners. We also discuss an experiment we designed and carried out to test the ideas presented here, along with the results and an analysis of our findings.

5.1 System Development

We have built a fully functional prototype of the aforementioned interaction system that implements many of the concepts it introduces. This system includes a working version of an underlying communication protocol based on Bluetooth Low Energy (BLE) and a corresponding Android app that makes use of this protocol to allow users to interact with autonomous robots.

One of the purposes behind building the system was to test its suitability for use by bystanders (or untrained people) in interacting with unfamiliar systems. Another reason was to investigate the how difficult implementing such a system might be and to identify potential complications with the design. Our goal is for the system to be flexible enough to convey a wide variety of information from different kinds of robots, but also structured enough to be practical to implement and easy for for people to use. In the following sections, we discuss the development of a Bluetooth communication protocol, Android user interface app, and the integration of our system into an autonomous robot.

5.1.1 Bluetooth Communication Protocol

One of the initial problems we needed to address was the process through which smartphones and robots could discover each other's presence. The Bluetooth V4.0 specification, also known as Bluetooth Low Energy (BLE), was well suited to perform this task as a relatively low range communication protocol (approximately 60m, unobstructed) that uses broadcast messages to allow peripheral devices to advertise their presence and the type of service(s) they provide. BLE is a commonly implemented communication protocol on smartphone devices, and can be easily incorporated into robotic systems. By having robots advertise themselves

using a well-known service universally unique identifier (UUID), smartphones can easily identify nearby robots (due to transmission range limitations) and distinguish them from other common BLE peripherals, such as heart rate monitors, wireless sensors, and proximity beacons.

BLE, as the name suggests, was designed for low power-draw applications and therefore is not optimized for high bandwidth data transfer. Unlike classic Bluetooth, BLE peripherals do not implement RFCOMM (a serial data stream protocol), but rather implements a General Attribute Profile (GATT) Server which defines and hosts Services and Characteristics that can be accessed by establishing a connection with the peripheral. As a result, BLE is well suited for device discovery but is not necessarily the best method of transferring large amounts of data between devices. Thus, while BLE can be used to implement an entire communication protocol, a more preferable method would involve using BLE advertisements to transmit information related to establishing connections over other communication channels such as classic Bluetooth, Wi-Fi Direct, or Wi-Fi over LAN. Nonetheless, having the ability to fall back on using a single protocol (despite it being relatively slow) was a compelling reason for us to initially implement our communication protocol using only BLE.

BLE specifies several different methods of accessing GATT attribute data from peripheral devices: read, write, notify, and indicate. Read and write are both initiated by the client (in this case, the smartphone), while notify and indicate are initiated by the server (the robot). Although notify and indicate can be used by the robot to push information to smartphones, the phone must have previously requested this information stream in order for data to be sent.

BLE allows for a maximum of 20 bytes to be sent per packet. To allow larger amounts of data to be sent between the devices, we developed a protocol for trans-

ferring data through groups of coded sets. Data is first separated into *messages*, or self-contained units of information. Each message is then divided into 20-byte chunks called *frames*, which fit into BLE packets.

Information can be sent reliably using data access methods that require confirmation (*read*, *write with reply*, and *indicate*); however, these calls result in significantly slower data transfer rates compared to their corresponding unreliable versions (*write without reply* and *notify*). To send data from the phone to the robot at a faster rate, the client periodically makes an initial *read* request from the server. The robot replies with the number of frames in its message, waits for the phone to subscribe to notify, and then begins sending the frames using the unreliable *notify* method. Unfortunately, this can result in missed frames. To combat this, frames are grouped into "batches" of 56 frames each and a sequence number corresponding to the frame's position within the batch are appended to the beginning of each Bluetooth packet. After receiving a batch of frames, the phone makes a *write* call to the robot, specifying any batch sequence numbers that it missed on the receiving side, or sends an empty packet if all of the frames were received. The robot then either resends the missing frames that were specified using *notify*, or begins sending the next batch of frames. Resent frames have a prefix bit added in front of the sequence number to distinguish resent packets that arrive with a delay from novel packets that might be in a subsequent batch. A similar process is employed for sending data from the phone back to the robot. The phone uses reliable *write* commands to specify to the robot how many frames it will deliver, followed by unreliable *writes* to transmit the data in batches. After each batch, the robot uses *indicate* to reliably reply with the sequence numbers of missing frames.

Another problem with Bluetooth is its inability to support many-to-many connections. This is due to the underlying design of Bluetooth piconets limiting the

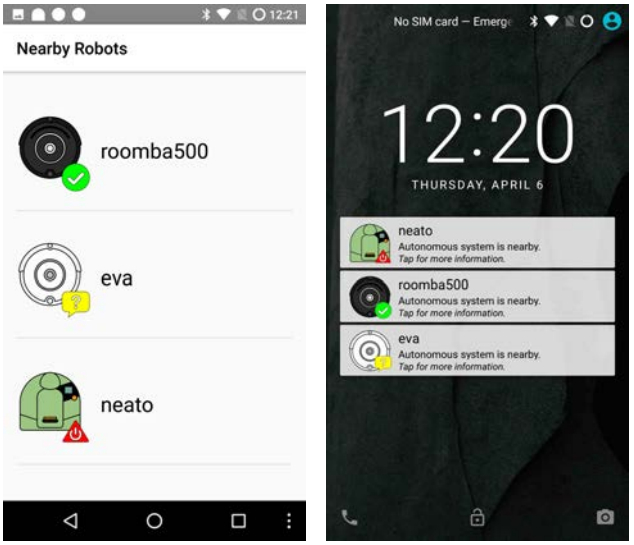
number of devices that can participate, and a previously established Bluetooth paradigm that allows a single host to connect to multiple peripherals, but restricting each peripheral to only one host at a time. In our configuration, the robots act as the peripheral devices, which effectively limits the number of smartphones that can communicate with the robot at any give time. To work around this limitation, we restrict the length of time clients can remain connected to a robot. This allows multiple smartphones to participate in a priority-based round-robin style connection, with priority given to people who are actively communicating with the robot.

To help improve the availability of information to users and decrease wait time, smartphones cache information from robots before it is actually requested by the user. The smartphones can then check to see if their information is outdated by comparing a cyclic redundancy check (CRC) value of their cached information against a published CRC value included in the robot's advertising data.

5.1.2 Android App: RobotLink

We created an Android app called RobotLink whose design was based on the user interactions concepts described at the beginning of this chapter. The architecture of the app used a model-view-controller that was implemented across three main software components: a background service (which implemented the Bluetooth communication protocol described in Section 5.1.1), a foreground app (the UI), and a notification system.

The controller was implemented as a background service that was responsible for receiving information over Bluetooth and caching it inside a data model used to provide information to the system's user interface running in a separate foreground app. Additionally, the data model was also monitored by a notification service



(a) Robot chooser (b) Background notifications

Figure 5.4: RobotLink nearby robot selection (*pull-style* interaction)

which managed the display of background notifications and popup messages. The model included any commands or response options that were specified by the robot which users could optionally use to communicate with the machine. Response options selected by users were passed back to the controller to be transmitted over the Bluetooth protocol.

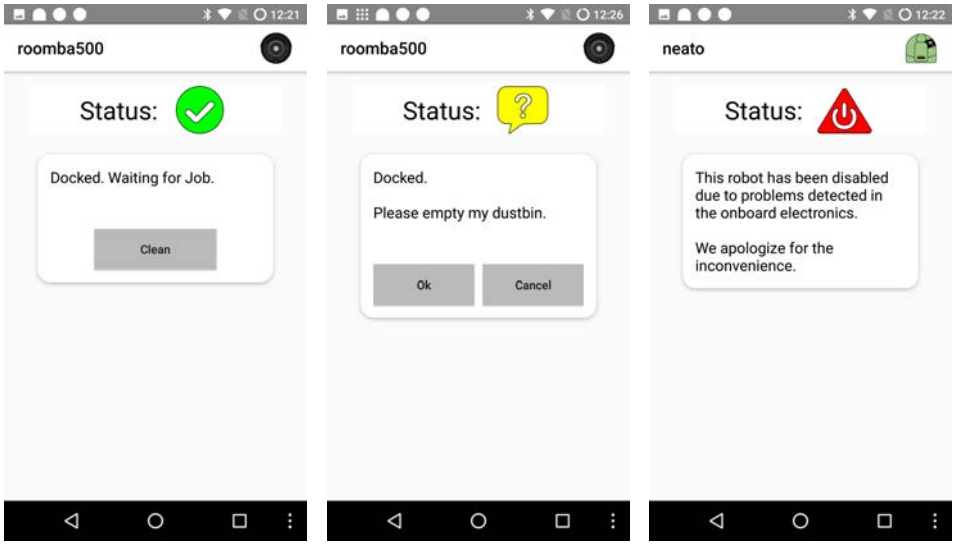
The background service was responsible for scanning for nearby robots, initiating periodic information transfers, keeping the cached model information up to date, and sending user input back to the appropriate robot. When the service saw an advertisement from the robot, it checked to see if the data cached in the model matched the most recent data on the robot by comparing CRC codes. If the service determined that it had outdated or updated data, it established a connection and either requested or transmitted data to the robot, respectively. Data received by the app was translated into data structures which were stored in the model for use by the notifications and the UI. If multiple robots were within range of

the smartphone, the background service tracked signals for each to monitor their continued presence. If the service determined that a robot had left the proximity of the smartphone, the robot was removed from the data model.

There was some difficulty in implementing the Bluetooth communication protocol using Android's Bluetooth API. Specifically, we encountered occasional problems with the "unknown" error code 133 while attempting to connect to robot GATT servers. This would occasionally result in Bluetooth ceasing to work on the phone. The problem was somewhat resolved by updating to Android 6.0 (Marshmallow). While this did not eliminate the 133 error messages, it did allow us to keep Bluetooth operational. Unfortunately, Android 6.0 also implemented a feature in which the device would assign itself a random MAC address while performing BLE advertisement scans, making it impossible to track individual phones from the robots' side.

5.1.2.1 Pull-style Interactions

The app accommodated both *pull-style* and *push-style* interactions. *Pull-style* interactions were achieved by either opening the app and selecting a robot from a "robot chooser" list (see Figure 5.4a), or by tapping on a background notification (see Figure 5.4b). Both methods listed robots by showing an iconified representation of the machines' physical appearances, a unique identification name which was also printed on the side of the robot, and a status icon. When robots moved out of range of the Bluetooth signal, they were removed from the data model. Subsequently, this resulted in them also being removed from the robot-chooser list and the removal of their background notification (if it had not already been dismissed).



(a) Ok (Docked) (b) Help request (c) Disabled

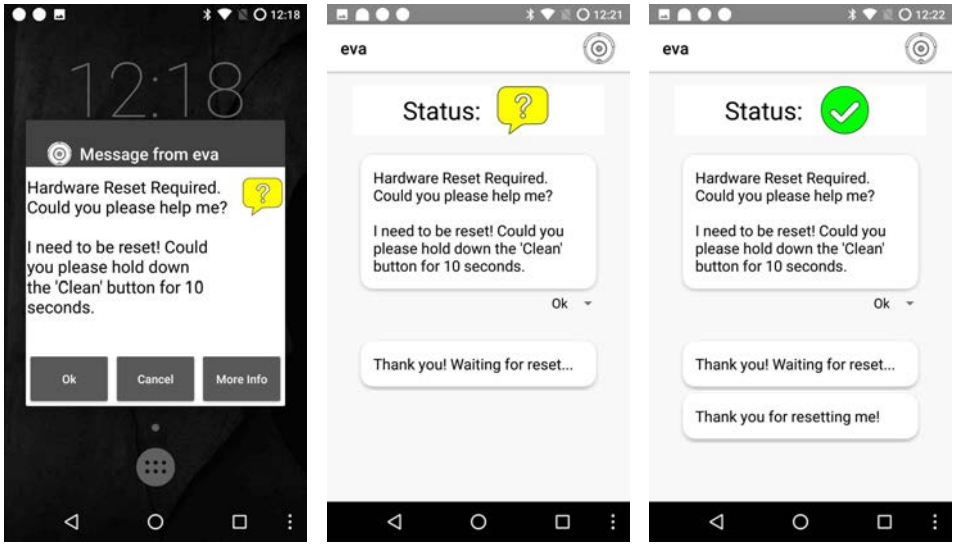
Figure 5.5: RobotLink status page examples

5.1.2.2 Status Pages

Once a robot was selected, the user was shown the robot's "status page" which contained additional information about the machine (see Figure 5.5). Maintaining cached information about all of the nearby robots in the data model improved the speed and responsiveness of the UI in delivering information to the user on these status pages. The top of the status page showed the robot's name and iconified picture, along with a simplified version of the robot's status in the form of a status icon (developed in Chapter 4). This was followed by a series of one or more progression boxes which described in more detail what the robot was doing and any options the user had for interacting with the machine, such as commands the robot was willing to perform or replies to messages and requests for help.

The status page was designed using several of the principles of Google's material design specification [Google, 2017]. For example, the app was designed using theme colors, and a "flat" design was used throughout the UI. Progressions on the

status pages were designed using "cards". Buttons on the status page made use of shadows to represent being raised or pressed.



(a) Popup notification (b) Status page help request (c) Help acknowledgement

Figure 5.6: RobotLink request to help reset a robot (*push-style* interaction)

5.1.2.3 Push-style Interactions

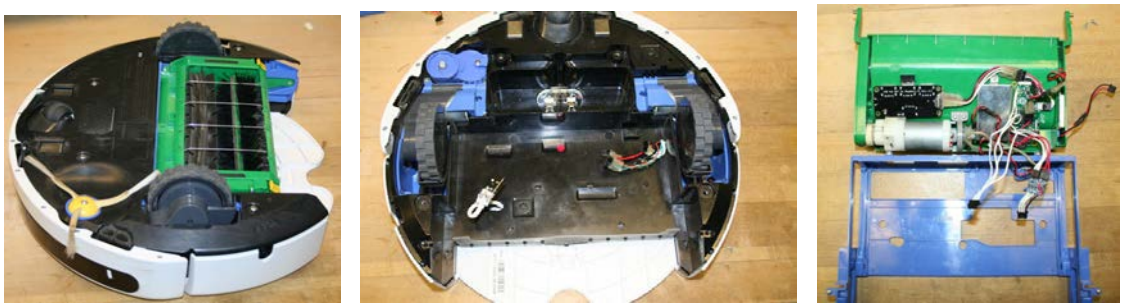
Changes to the model could trigger *push-style* interactions in the form of pop-up notification alerting the user to a message sent by the robot (see Figure 5.6a). The information seen in the pop-up message would also become available on the robot's status page (Figure 5.6b), along with any response the user may have provided. After the *push-style* interaction, subsequent messages sent from the robot to the person (especially those related to the information in the pop-up) were typically posted directly on the status page rather than as additional pop-up messages (Figure 5.6c), since the user's attention had already been captured.

Figure 5.7: Robot vacuums modified to work with our smartphone app, RobotLink.

5.1.3 Robot Hardware

We implemented our Bluetooth communication system on five robot vacuum cleaners (see Figure 5.7), including several models of iRobot Roombas and a Neato XV-11 which would later be used as part of an experiment described in Section 5.2. The upgrades were implemented so there were no visible modifications to the robots' external appearances. Additional electronics that were added to the robots (described below) shared the same power source as the rest of the robot, eliminating the need for additional or modified charging systems. By default, the new electronics did not interfere with the robots' normal operation, allowing us to preserve the original interface and user interaction design by the manufacturer when desired.

Most of the new electronics were stored inside the sweeper assembly on the Roombas (see Figure 5.8), or were fit into the front right corner beside the dustbin on the Neato XV-11. Placing the majority of the electronics inside the Roomba's sweeper assembly allowed most of the system to be removed without needing to completely disassemble the robot. Small connectors bridged the electronics in the sweeper assembly with connection points in the rest of the robot, allowing the assembly to be easily separated from the main body for maintenance and testing.



(a) Bottom with sweeper (b) Sweeper assembly removed (c) Inside assembly

Figure 5.8: Roomba electronics in sweeper assembly

Each robot received an additional single board computer, dual band WiFi radio, a specialized Bluetooth Low Energy radio, powered USB hub, serial interface circuit, one or more additional sensors, and switching voltage regulator (see Figure 5.10). The single board computer used was a Raspberry Pi (RPi) Zero running a 1GHz single core ARM processor with 512MB of RAM, and ran all of the additional software needed for remotely controlling the robot over WiFi, interface with the RobotLink smartphone app over Bluetooth, and performing data logging (Figure 5.9A). A hardware reset button for the RPi was hidden deep inside the robots' wheel wells to allow us to reboot the machine without disassembly. Panda Wireless N600 dual band WiFi adapters were selected for their Ralink RT5572 chipsets (which are well supported by Linux), relatively small size, and good signal quality. WiFi was used to provide backchannel control and communication to each robot (Figure 5.11A).

In order to bypass the Linux kernel's Bluetooth stack and gain greater control over the behavior of the Bluetooth radio (including the ability to run a GATT server), we used Bluegiga BLED112 USB Bluetooth Low Energy development dongles (Figure 5.9C). The firmware of the BLED112 could be re-flashed to modify it to behave as a customized embedded GATT server, which could be further con-

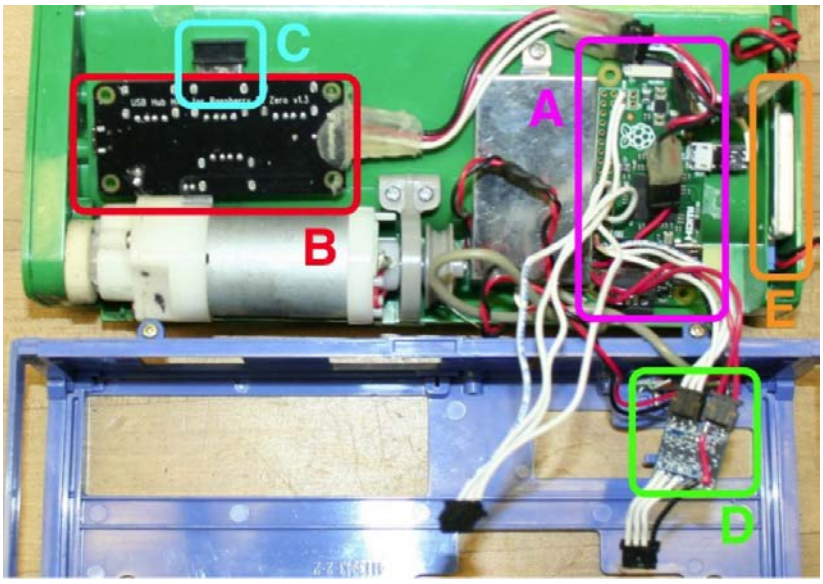


Figure 5.9: Sweeper assembly electronics details. (A) Raspberry Pi Zero ARM computer (B) Powered USB Hub, (C) Bluegiga BLED112 USB Bluetooth dongle (D) Serial interface logic level converter (E) Universal Battery Elimination Circuit (UBEC) 5V 3A DC/DC buck converter.

trolled via a serial interface using a specialized protocol called BGAPI. Both the Bluetooth and WiFi hardware were USB devices, so a powered USB hub was added to interface them with the RPi (Figure 5.9A).

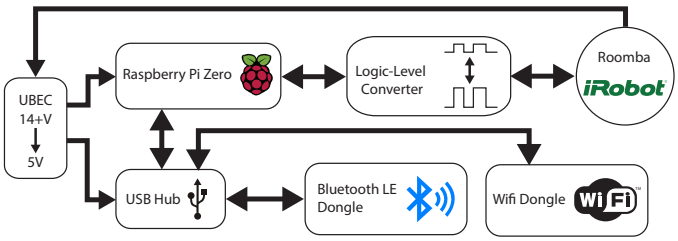


Figure 5.10: Robot hardware block diagram

While the Neato XV-11 ships with a built-in USB interface, the Roomba uses a 5V TTL serial connection. Additionally, the Roomba features a keep-alive pin which needs to be toggled occasionally in order to prevent the serial interface from falling asleep. This required us to add a logic level converter circuit (Figure 5.9D) to the Roombas to connect them with the RPi's 3.3V GPIO and UART pins

(Figure 5.11B). Most of the robots were unable to detect whether or not their dustbin was present or had been removed, an attribute we wished to track. We were able to accomplish this by fitting small magnets to the dustbin containers and placing Phidgets #3560 magnetic contact switches inside the robot (Figure 5.11C).

Finally, all of the robots were powered by 14.4V batteries while all of the additional electronics required 5V power. To provided power to the added components, we used a device called a "Universal Battery Elimination Circuit" (or UBEC, see Figure 5.9E) frequently used in RC hobby airplanes and helicopters which implements a 5V 3A DC/DC buck converter (step-down switching regulator).

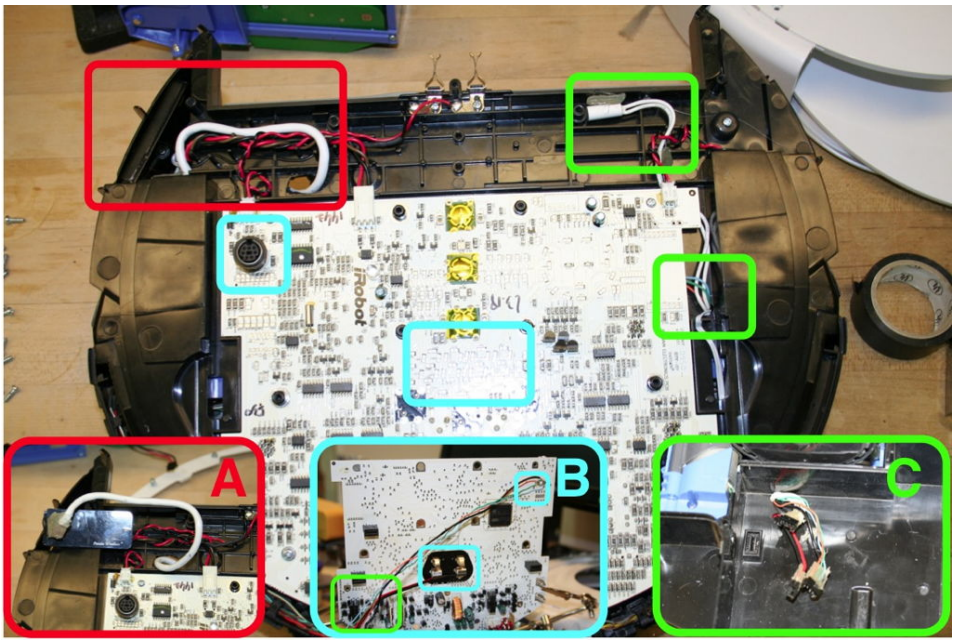


Figure 5.11: Electronics inside the main body of the robots. (*A*) Wifi adapter placement, (*B*) Serial and power connections, (*C*) Wiring connectors from the mainboard, a Phidgets magnetic switch (dustbin sensor), and reset button to the sweeper assembly.

5.1.4 Robot Software

All of the software for controlling each robot ran on the Raspberry Pi Zero single board computer embedded inside each robot. Each system was loaded with identical copies of the Linux based Raspbian operating system (a derivative of Debian Jessie), a manually compiled ARM version of ROS Indigo (base dependencies only), wi-cd network management software, and SupervisorD for application process management and control over XML-RPC.

The robots ran three main software applications developed specifically for this project - the robot control software, a software update system, and a host telemetry server. The robot control software (or controller) was used during the experiment and integrated the robot specific hardware interface, Bluetooth interface, and experiment interface in order to dictate how the robot would behave at any given moment. The software update system provided a RESTful web service that allowed us to push code updates to individual robots or simultaneously to all robots without needing to be logged in over SSH. The host telemetry servers allowed us to remotely monitor the health of all of the robots used in the experiment by providing information about the host systems, including CPU load, memory usage, system uptime, and networking statistics such as latency and signal quality.

5.1.4.1 Controller Software

The controller software was responsible for dictating the robot's behavior. To minimize the amount of duplicate code that needed to be maintained, all of the robots used a single generic controller with software routines that were shared by all the machines. As part of the bootstrapping process, the controller would load machine specific settings and software modules defined by a config file whose name matched the robot's hostname. This allowed all the robots to share code repositories and

enabled us to quickly make changes to multiple systems via our software update system. The remainder of the controller's functionality was divided up by the three subsystems which it managed - the hardware interface, the Bluetooth interface, and the experiment interface.

5.1.4.2 Hardware Interface

The robot hardware interface provided an abstraction layer between the robot hardware and the controller software. All of the robot manufacturers provided an API to the built-in electronics that was accessible via serial connection. The hardware interface managed this connection, tracked sensor values and hardware state, and relayed commands to the hardware. In addition to the serial API, the hardware interface also used the Raspberry Pi's GPIO to interface with additional sensors that were added to the robots. This interface was primarily used for tracking the state of the dustbin and to generate a keep alive signal for the Roombas.

The Roomba hardware interface was based on iRobot's Open Interface (OI) specification and a ROS driver for the Turtlebot designed for teleoperation. The Turtlebot driver allowed the robot to be directly controlled by placing the robot hardware in a state called manual mode which overrode all the vacuum's default behaviors. Unfortunately, our use case for the robot was actually to have people using the vacuum the way it was intended to be used. This meant that for us to use the robot in manual mode we would have had to re-implement all of the robot's behaviors ourselves, including the button interface, random wander, find-dock, and even battery charging.

Instead, we developed a new driver for the Roomba that leveraged a second hardware state called passive mode in addition to manual mode. Passive mode allowed us to use the robot's default behaviors while still being able to read sensor

values and control the robot through emulated hardware button presses. Custom behaviors such as simulated error states (e.g. the robot needing reset) were accomplished by temporarily switching the robot into manual mode, while normal robot operation, via either the onboard buttons or remotely through emulated button presses, were managed in passive mode.

There were two main problems with using passive mode to control the Roomba. The first problem was that despite exposing all of the onboard sensors through the OI API, the robot's state (e.g. idle, cleaning, looking for dock, etc) was not provided. Since some of our functionality depended on knowing the robot's state, we needed to implicitly derive this information from sensor information and track the state in our software. State estimation was accomplished using a combination of robot sensor data (e.g. whether the dock was detected or the robot had been lifted up), monitoring button presses, and watching for other indicators to confirm state changes (such as the sweeper brush starting or the battery entering charging mode).

The second problem was that it was not possible to differentiate between physical button presses on the robot and emulated button presses (which happen to be the only mechanism available for programmatically engaging various robot behaviors). Therefore, we needed to be capable of logging the differences between these events as part of data collection. Whenever a button was pressed on the robot it was reported through the serial interface. Unfortunately, emulated button presses sent through the API also appeared as physical button presses. This issue was resolved by explicitly tracking emulated button presses and modifying the button press data returned by our drivers to not include the associated button press event that occurred immediately after the emulated press was sent.

Several other hardware interfaces were developed in addition to the Roomba hardware interface for this project. Two versions of a hardware interface were

developed for the Neato XV-11. The first interface was developed using C++ in an attempt to create a ROS node that would run fast enough on the RPi Zero to achieve position localization by publishing the Neato's laser values and wheel odometry data to a node running Adaptive Monte Carlo Localization (AMCL). These efforts were abandoned after it was decided that the Neato robot would only be playing the role of a "disabled" robots that would not be moving. The replacement Neato interface was designed to explicitly prevent users from being able to make the robot start cleaning, while logging all user interactions including button presses, lifting the robot, placing the robot on its charger, and removing the dustbin.

A second version of the Roomba interface was also developed, designed for use by two older model Roombas which only implemented a small subset of iRobot's OI API. Development of this hardware interface was cancelled after we determined that our hardware modifications were causing battery stability problems that were likely to later interfere with data collection. As a result, one of these robots was replaced with a newer model while the other was modified to use an alternative power source and have the robot's sensors directly connected to the Raspberry Pi. The final hardware interface was designed to work with the Stage robot simulator software during early development, prior to the completion of any of the robot hardware.

5.1.4.3 Bluetooth Interface

The controller software ran a server implementation of the Bluetooth protocol described in Section 5.1.1, which was used to communicate with the RobotLink smartphone app. The server API allowed the controller to set the robot's identification name, specify a platform type (conveyed using an iconified representation of the robot), state information (conveyed using a robot state icon), and progres-

sions. Progressions were implemented as an ordered list of "messages." Each message consisted of formatted text, optionally attached media resources (such as images), an optional list of commands or responses a user could choose to send back, and whether or not the message should be treated as a *push-style* interaction and displayed as a popup. This information was organized according to JSON schema definitions and serialized for transmission.

We decided to have our robots offer users the same controls though the Bluetooth interface as the physical button controls found on the robots. The status information sent to users was derived from either the realtime hardware state, or from simulated problems generated by the experiment interface. The Bluetooth radio could be toggled on and off, allowing the robots to hide their presence and effectively disable remote control via the smartphone app.

5.1.4.4 Experiment Interface

The controller software interfaced with the experiment manager software using ROS. The experiment manager specified when the robots should be enabled or disabled, what behaviors they should exhibit, and facilitated a logging system for recording data (see Section 5.2.4.1 for more details). A specialized ROS adapter called "ReliableNode" was written to allow the robots to operate whether or not ROS happened to be running. This allowed us to start the robots' controllers and leave them running despite shutting down and restarting the other ROS enabled components of the experiment. The experiment interface was capable of generating simulated problems with the hardware on demand in order to allow us to investigate people's responses to unexpected behaviors in a reproducible manner.

5.2 Experiment

The experiment described in this section represents our initial investigation into the use of smartphones as a platform for establishing a ubiquitous communication paradigm designed to allow untrained people, such as bystanders, to gain basic information from and interact with a wide range of autonomous robots. In this experiment, we asked participants to perform two simultaneous tasks - manage a fleet of vacuum cleaning robots in "cleaning" an area of a floor while playing a video game on a smartphone at the same time. Our objective was to gain insight into whether participants would be able to effectively use our smartphone app, RobotLink, to interact with a number of autonomous robots with varying levels of functional ability without any prior training, and gather information about their opinion of our system. The experiment used a within-subjects design where each participant had the opportunity to control the robots using only the manufacturer's interfaces during one run, and both the manufacturer's interface and our RobotLink app during another. Our hypotheses in this study were as follows:

Hypothesis 5.1: *Participants would be able to determine which robot they were communicating with using our smartphone based system, despite similarities between robots.*

Hypothesis 5.2: *Using our system, participants would be able to retrieve information about the robots they were working with, identify solutions to problems faster, and allocate their time more appropriately compared to the default interfaces.*

Hypothesis 5.3: *Participants would prefer having access to the additional information provided by our system over the default manufacturer interface alone.*

(a) Room Layout (b) Person in game-playing zone (c) Interacting with a robot

Figure 5.12: The setup of the room in which the experiment took place.

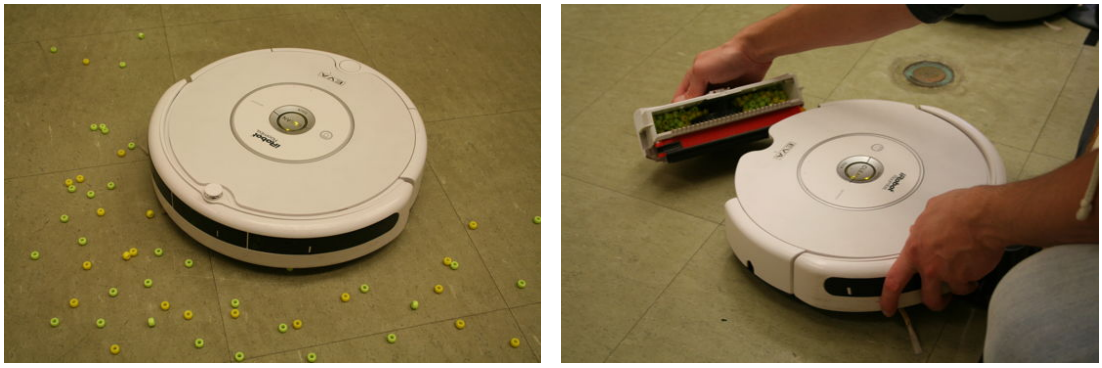
5.2.1 Task Description

Participants were asked to perform two simultaneous tasks: use a set of three robots to collect plastic beads scattered around on the floor and play a simple video game on a smartphone. The video game task was a simple, skill-less game of balloon popping which participants played on a smartphone we provided for them. Animated balloons drifted up from the bottom of the screen to the top and "popped" when touched, earning the participant points (see Section 5.2.4.2). The game required the participants to pay attention, as some of the balloons were marked with skull and cross-bones symbols and would take away earned points if popped. The second task asked participants to use 3 apparently commercial-off-the-shelf (COTS) robot vacuums to collect small plastic beads scattered around the floor inside an area fenced off by 2x4 pieces of wood called the "Cleaning Zone". Participants were required to remain outside of this area and were not permitted to collect the beads themselves, forcing them to use the robots to accomplish the task. Finally, participants were asked to take a number of questionnaires, including both pre- and post-experiment questionnaires, as well as a post-run questionnaire after each run.

In order to track the amount of time participants spent physically interacting

with or observing the robots, these activities were designed to be mutually exclusive from time the participant spent playing the Balloons game. This was accomplished by only allowing the game to be played when the participant was inside a specially marked off area called the game-playing zone which had an obstructed view of the robots (see Figure 5.12). When participants left the game zone, the Balloons game interface became disabled, preventing people from either gaining or losing points in the game until they returned (see Figure 5.16b).

Participants were given six and a half minutes to both play the Balloons game and use the robots to collect beads. Participants were responsible for keeping track of the time remaining by using either a digital clock showing the game time that was positioned inside the game-playing zone, or by using the game timer inside the Balloons game (which also showed the experiment time). While they earned money based on the number of points that were scored in the video game, participants only got to keep the fraction of those points which corresponded to the percentage of beads the robots were able to collect (i.e. if they collected 70% of the beads they would get to keep 70% of the points scored in the game). Prior to the start of each run, a set amount of beads (approximately 100g) was measured out and scattered across the floor inside the cleaning zone. At the end of each run, the beads collected by the robots were measured by weight to calculate the percentage collected, while the remaining beads were removed from the cleaning area by the experimenter. Participants were also instructed that they would lose an additional 100 points (approximately 45 seconds worth of Balloons game playtime) for each robot that was not on a charging station when time ran out. Participants had two chances (or runs) in which to perform these tasks, and received compensation based on the higher of their two final scores. Each person received $5 for simply completing the study, and could earn up to $10 more based on their performance.



(a) Robot collecting beads (b) Emptying a dustbin

Figure 5.13: Robot bead collection during the experiment.

A copy of the script the experimenter read to participants during each session can be found in Appendix A.1.

5.2.2 Independent Variables

This experiment used a within-subjects (repeated measures) design in which each participant performed two runs. Participants were assigned into one of four experimental conditions corresponding to the two independent variables in the experiment: the order in which the RobotLink app support was used in the runs and the order in which two different starting configurations were used. Thus, each participant experienced one run with the RobotLink app enabled and one run with only the manufacturers' interfaces. Between runs, the experimenter replaced two of the three robots used in the previous run in full view of the participant as part of "resetting the task," while the participant filled out the post-run questionnaire. Prior to the start of each run, the experimenter explained whether or not the participant would have access to the RobotLink app during that run. The RobotLink app allowed participants to retrieve information about each of the three robot's status, allowed robots to request help from participants, and allowed participants

Cond	Phone Support	Run	Robot Starting Conditions R_A	R_B	R_C	R_D	R_E
1	Enabled	1	Easy	Dead	Help		
1	Disabled	2	Help			Easy	Dead
2	Enabled	1	Easy			Help	Dead
2	Disabled	2	Help	Dead	Easy		
3	Disabled	1	Easy	Dead	Help		
3	Enabled	2	Help			Easy	Dead
4	Disabled	1	Easy			Help	Dead
4	Enabled	2	Help	Dead	Easy		

Table 5.1: Experiment conditions. During each run, one robot was "easy" to start, one required help (from the participant) before it would start running, and one was "dead" and never would start working.

to remotely send commands to the robot (all of the commands available were are also available using the controls present on the physical robot.)

The experiment made use of 5 robot vacuum cleaners: four iRobot Roombas and one Neato XV11. The four iRobot Roombas consisted of two working Roomba 500 models (R_A and R_D), a working Roomba 600 model (R_C), and a non-working Roomba Discovery model (R_B). The Neato XV11 (R_E) was intentionally programed not to work. Three of these vacuums were used during the first run, after which two of the three were replaced before starting the next run. There were two combinations of robots that were switched between, each of which consists of two "working" robots and one "non-working" robot. Specifically, *Group 1* consisted of R_A, R_B and R_C while *Group 2* consisted of R_A, R_D, and R_E.

The "working" robots all performed their default behaviors (as specified by the manufacturer), except that the length of time they ran for was artificially shortened, and some "problems" were artificially introduced. During each run, each of the three robots exhibited a different level of functionality: one robot was "easy" to start, simply requiring the push of a button, one required "help" from

the participant before it would start running (it needed to be reset), and the last one played "dead" and would never start working (see Table 5.1). Two minutes after a robot started cleaning, it would automatically start returning to its dock. After the "easy" robot returned, it would require the participant to come to empty its dustbin before it would be able to start cleaning again. On the other hand, the "help" robot could immediately be told to resume cleaning (even before it finished returning to its dock) when its two minutes were up, and never required its dustbin to be emptied.

The robot that needed to be reset and robot that needed its dustbin emptied would signal users they needed attention using audio (beeps) and visual (blinking lights) indicators. The robot that needed to be reset would flash a red LED ring around the power button, illuminate a red error symbol in the shape of a circle with an exclamation mark in the center of it, and would periodically play a distinct error tone until the participant pressed and held down on the power button. During runs in which the RobotLink app was enabled, this robot would use a *push-style* interaction to trigger a popup message to appear on the participant's smartphone. The reset events would always take place at the beginning of each run. The robot that needed its dustbin emptied would illuminate a yellow LED ring around the power button, flash a blue LED labeled "dirt detect", and would occasionally play an different error tone until the dustbin was removed. During the run in which the RobotLink was enabled, participants would be able to view information about the dustbin needing to be emptied by viewing that robot's status page (a *pull-style* interaction). Time spent trying to make the third robot work was wasted. One of the dead robots had no lights on and showed no indication it even had power. The other dead robot had a single LED light that was turned on, but did not display anything on a built-in LCD screen and showed no other signs to indicate it might

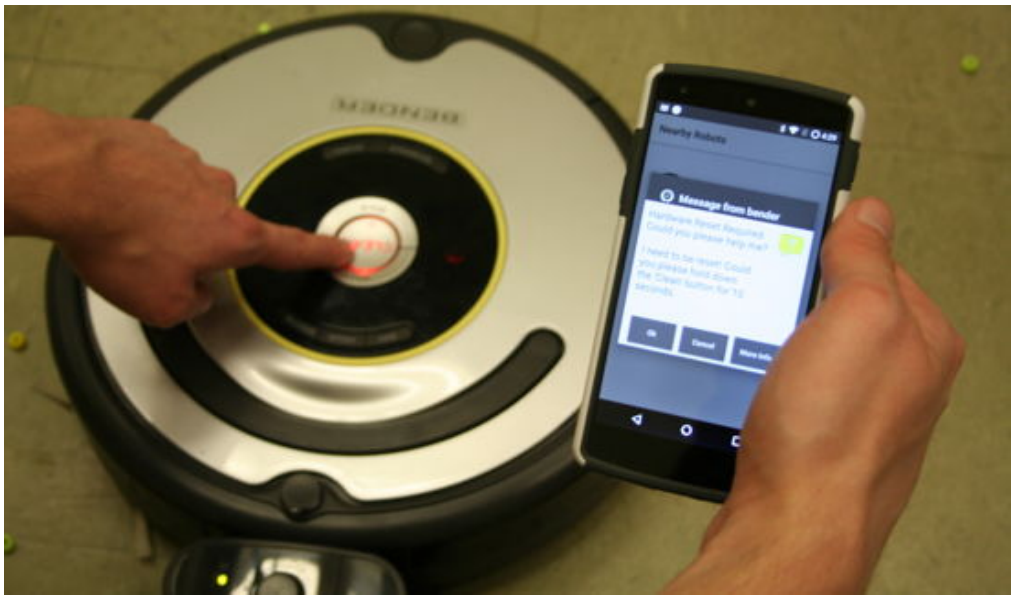


Figure 5.14: Participant using the RobotLink app to reset a robot.

be able work. The dead robot could also be viewed using *pull-style* interactions when the RobotLink app was enabled.

5.2.3 Dependent Variables

We collected data from several different sources, including pre- and post-experiment questionnaires, post-run questionnaires, data logged both on the robots and the smartphone, manual annotations made by the experimenters, and video recordings. Questionnaire data was collected using Qualitrics survey software. A copy of the survey questions used can be found in Appendix A.2. The following variables were measured by our post-run surveys:

- participant workload using NASA TLX,
- participant confidence in the smartphone system, and
- whether participants understood where data on the phone was coming from.

The following variables were recorded by the experimenter at the end of each run:

- whether or not all the robots were back on their chargers when time expired,
- the percentage of beads collected (measured using a scale), and
- the game score on the phone.

The following variables were measured from the smartphone in all conditions:

- the amount of time the participant spent playing the video game,
- the participant's final score when time expired,
- the number of times the participant started and stopped playing the game when leaving the game zone,
- the number of times the participant started and stopped playing the game without leaving the game zone, and
- the rate of participant's game score increase over time.

The following variables were measured from the Android app (in the smartphone condition only):

- number of times the participant accessed each robot's app page via the pull style notifications,
- amount of time participant spent viewing each robot's app page (phone turned on, switched between pages, etc),
- number of times participant looked at background notifications,
- whether the participant explicitly agreed or declined (via the app dialog buttons) to help the robot,

- whether the participant agreed to help the robot, accessed more information, or declined, via the push dialog, and
- whether the participant issued commands to the robots to begin cleaning, pause, or return to their charging station.

5.2.4 Experiment Development

Several pieces of software infrastructure were developed for this experiment, including the Balloons game, a wall clock, an automated detection system to determine when the participant was inside the game-playing zone, a data logging system, and several tools to help experiments control, monitor, and annotate experiment progress. The experiment itself was developed using ROS to coordinate and synchronize operation of the various components which were running across 10 different machines and devices. In addition to the 5 robots, the experiment also used 3 computers (an experiment control station, a machine on the desk in the game-playing zone, and a remote machine setup for a dedicated observing experimenter) and two Android devices (a phone used by participants and a tablet used by the remote experimenter for manual annotation).

5.2.4.1 Experiment Manager and Logging System

The experiment was controlled using a centralized coordination system called the Experiment Manager. This software broadcast the experiment variables being used for a run to the rest of the system, monitored and assigned robot behaviors, tracked and published the remaining time during each run, managed the logging system, and started and stopped the applications that were run on the machine located in the game-playing zone (such as questionnaires and the game clock). The logging

system was created using ROS and consisted of a specialized node that could log all published topic data into bag files into a specified file location. Data logged in this manner was automatically time synchronized and could be replayed or mined at a later time to extract detailed information about what occurred during each run of the experiment. Robots, cameras, the game-playing zone LIDAR, audio recording system, Balloon game, and Android tablet annotation app were all designed to publish logging data over ROS to be recorded by this system.

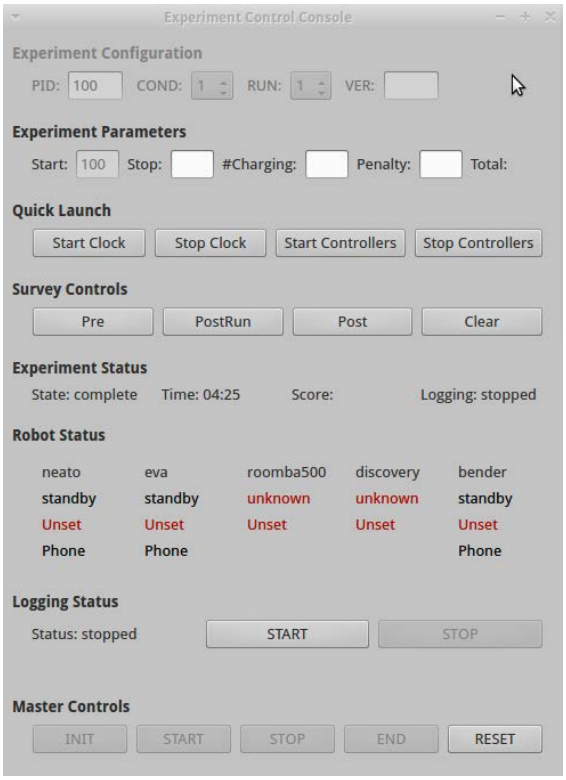
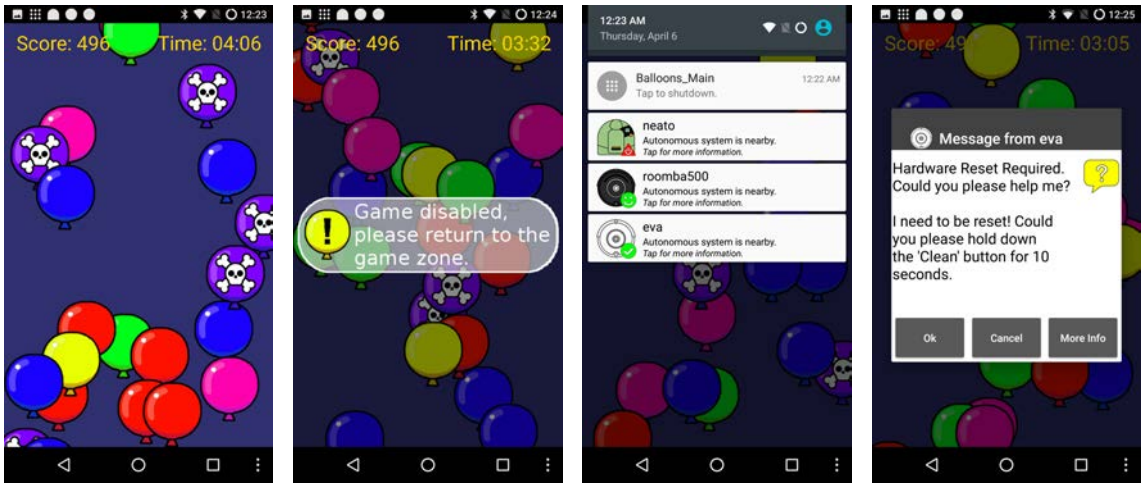


Figure 5.15: Experiment control panel

5.2.4.2 Balloons Game

The Balloons game was developed as a secondary activity to prevent participants from continuously monitoring the robots for the entire length of the experiment. The concept of the game was based off a fictional game known as "Jerry's game"

from the Adultswim show *Rick and Morty*, and modified for use in our experiment. Participants received points for popping balloons, but lost points for popping balloons with the skull and crossbones (the good balloons made a different popping sound from the bad balloons to help users identify when they made a mistake). Our game featured a static "difficulty" level which did not vary over time and could not be "lost" or otherwise ended prior to the end of each run (it was not possible to score fewer than 0 points). The game was directly integrated with the rest of the experiment using ROSJava; it started and ended in synchronization with each run, and the countdown timer in the upper right-hand corner directly reflected the time remaining in the run. Additionally, the game monitored information published by the game-playing zone detection algorithm (see Section 5.2.4.3) in order to automatically disable user input from the game and display a warning message whenever participants exited the game-playing zone (see Figure 5.16b). The current game score, information about whether or not the game was being displayed on the screen, and whether or not background notifications were being viewed were all published and logged over ROS topics.



(a) Normal game play (b) Disabled (c) *Pull-style* interaction (d) *Push-style* interaction

Figure 5.16: Balloons game being used during the experiment.

5.2.4.3 Game-playing Zone Detection

The experiment design required that we accurately track when people exited and entered the game-playing zone. This was accomplished using an Hokuyo UTM-30LX LIDAR to track peoples position relative to the known coordinates of the space. Data from the LIDAR was transformed from polar coordinates into a cartesian system relative to the game-playing zone using ROS' built-in TF transform library. The transformed points were then tested for inclusion inside the game-playing zone using MatPlotLib's built-in *Path* class which features a multiple point-in-polygon testing algorithm.

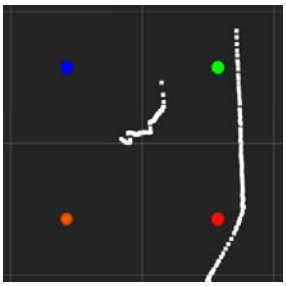
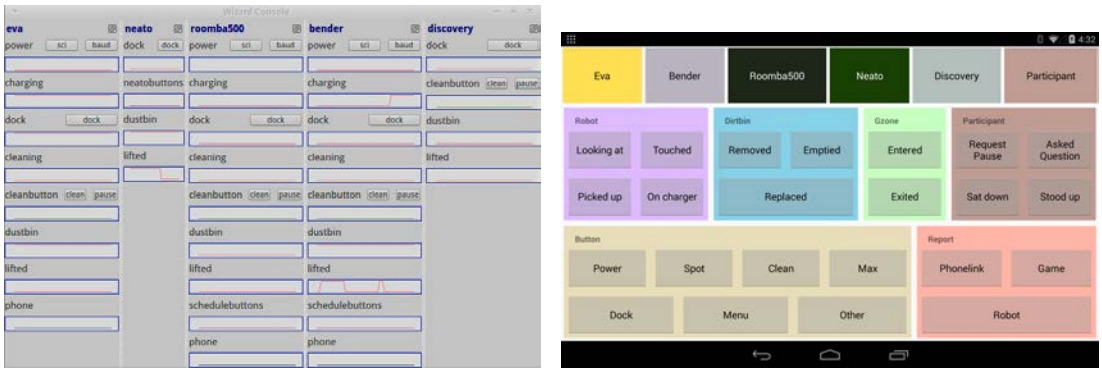


Figure 5.17: Visualization of a participant standing inside the game-playing zone

5.2.4.4 Wizard Interface and Annotation App

Two tools were developed to aid a remote experimenter (known as the "wizard", despite their not actually needing to perform any wizard-of-oz duties), whose job was to observe and annotate the experiment sessions. The *Wizard Interface* (Figure 5.18a) consisted of a series of plotted lines showing changes in values for various sensors onboard the robots, which would result in sharply slanting lines inside the plots to help the wizard confirm that events onboard the robots were functioning properly and that events were being tracked by the logging system. Earlier versions of this software permitted the wizard to remotely control a variety of robot behaviors, while the final version only allowed the wizard a limited set of commands that



(a) Wizard Console (b) Annotation app

Figure 5.18: Interfaces used by the experiment wizard

were necessary to manually recover robots from infrequently encountered edge-case conditions. The other tool developed for the wizard was an Android app designed to be run on a tablet and used for manually annotating events during the experiment that could be logged over ROS. The annotation app worked by selecting a "subject" from the top row, followed by a "verb" which defined the event being logged (see Figure 5.18b).

5.2.4.5 Robot Monitor

A robot health monitoring tool (Figure 5.19) was created early during development to aid in debugging hardware, software, and network related problems on each of the robot systems being developed. The robot monitor connected to the host telemetry server being run by each system using XMLRPC to query information about each machine's CPU load, memory usage, uptime, and network latency. It could also be used to start and stop an individual robot's controller software, and reboot machines if necessary. It was later used during the data collection process to help experimenters quickly confirm whether any of the robot hardware systems were experiencing problems.

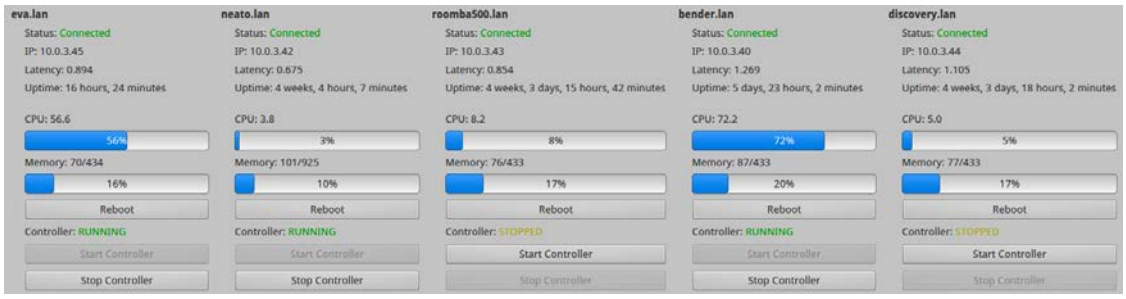


Figure 5.19: Robot health monitor system

5.3 Results

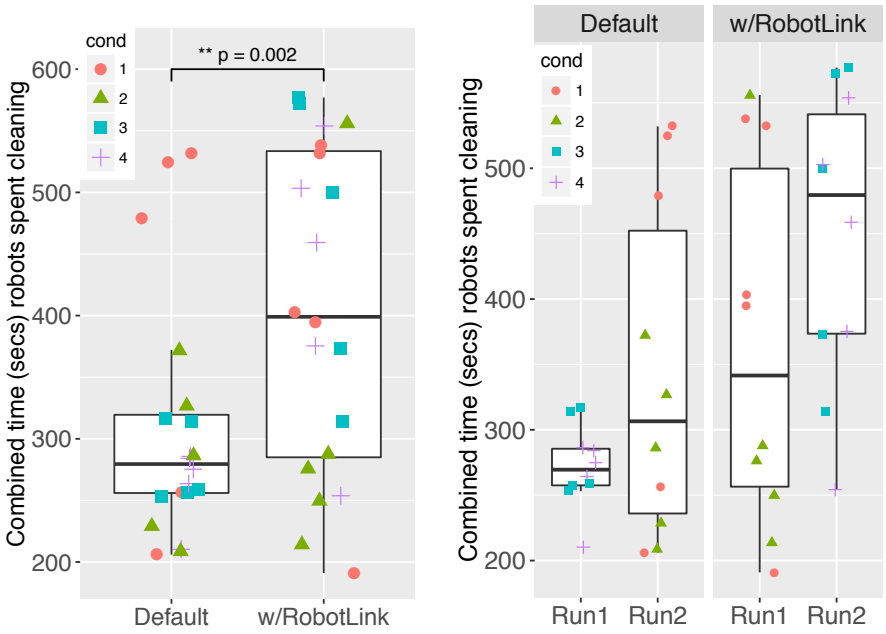
Twenty people (14 men, 6 women) between the ages of 18 and 33 participated in our experiment. All 20 participants had previous experience with smartphones - seven with Android, seven with iPhone, and six with both. One person also had experience with a Windows phone and three had experience with Blackberry devices. Two people had prior experience using iRobot Roombas and one person had previously used a Philips robot vacuum. However, the remaining seventeen participants had no prior experience with robot vacuum cleaners.

The majority of our analysis consisted of a series of 2x2 mixed-groups factorial ANOVA that were performed to examine the effects of participants' use of the RobotLink app and their assigned experiment condition on a number of dependent variables, including the combined amount of time each robot spent cleaning, the number of times participants pressed buttons on the robots, participants' scores and number of penalties in the Balloons game, time spent playing the Balloons game, time spent outside the game-playing zone, and participant workload.

5.3.1 Robot Usage

Runtime: There was a significant main effect of having access to the RobotLink app on the total combined amount of time the robots spent cleaning [$F(1, 16) =$

$17.052, p < 0.001$] (see Figure 5.20a). There was a significant interaction between the experiment condition and the presence of the phone [$F(3, 16) = 3.49, p = 0.04$], however there was no main effect of the experiment condition by itself [$F(3, 16) = 1.064, p = 0.39$] (see Figure 5.20b). A post-hoc two-tailed t-test showed that the combined time that robots spent cleaning was significantly higher during the run in which people had access to the RobotLink app ($M = 406, SD = 131$) compared to the run in which it was disabled ($M = 307, SD = 98$); $t(19) = 3.49, p = 0.002$. In other words, participants were able to keep the robots cleaning for a longer period of time by using the RobotLink app. While this was especially true for people who used the app during the second run, it was also generally true for the people who used the app in the first run.



(a) by RobotLink app (b) by run number

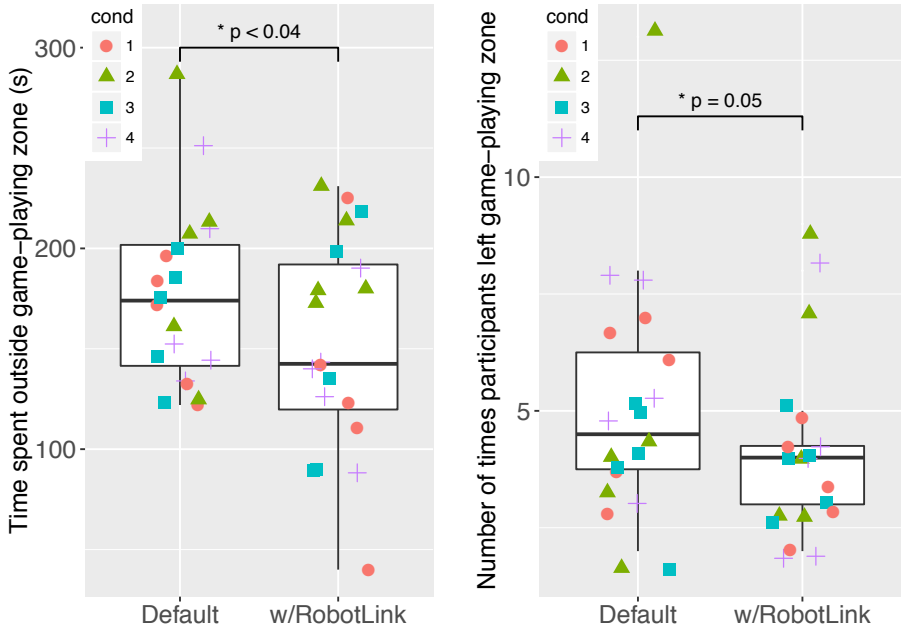
Figure 5.20: Time robots spent cleaning

Number of working robots: The number of robots participants were able to get working during the run in which the RobotLink app was enabled was compared with the run in which the app was disabled using McNemar's test (which is similar to a chi-squared test, but for paired data). The results showed there were significant differences in the number of robots participants were able to get working ($p = 0.004$). Participants were more likely to get two robots working while using the app (70%) than without the app (20%). No one (0/10) was able to get both robots working during their first run without using the RobotLink app. However, 6/10 people were able to get both robots working in the first run with the app, and 5/10 were able to get them both working without the app during their second run. Everyone (10/10) who had the app during the second run was able to get both robots working. This indicates that the app provided participants with the information necessary to get a second robot working.

5.3.2 Game-playing Zone

Time outside game-playing zone: There was a significant main effect of having access to the RobotLink app on the amount of time participants spent outside the game-playing zone [$F(1, 16) = 4.589, p = 0.048$] (see Figure 5.21a). Participants could only play the Balloons game while inside the game-playing zone, but had to leave it in order to physically interact with or monitor the robots. There was no main effect of the experiment condition variable [$F(3, 16) = 1.58, p = 0.23$] and there was no interaction between the app and the condition [$F(3, 16) = 0.531, p = 0.66$] on time spent outside the game-playing zone. A post-hoc two-tailed t-test showed that the time participants spent outside the game-playing zone observing and interacting with robots was significantly less during the run in which they had access to the RobotLink app (M=151, SD=53) compared to the run in which it

was disabled ($M = 176,\ SD = 44$);$t(19) = -2.23, p = 0.038$. This means that participants spent less time watching and physically interacting with the robots when they used the RobotLink app.



(a) Time outside

(b) Number of switches

Figure 5.21: Game-playing Zone

Context switches: There was a significant main effect of having access to the RobotLink app on the number of times participants switched between being inside and outside the game-playing zone [$F(1, 16) = 4.47, p = 0.05$] (see Figure 5.21b). There was no main effect of the experiment condition [$F(3, 16) = 0.36, p = 0.77$] and there was no interaction between the app and the experiment condition [$F(3, 16) = 1.221, p = 0.33$] on the number of times participants switched between being inside and outside the game-playing zone. A post-hoc two-tailed t-test showed that the number of times participants switched between the game-playing zone was significantly less during the run when they had access to

the RobotLink app ($M = 4.1, SD = 1.92$) compared to the run in which it was disabled ($M = 5.1, SD = 2.57$); $t(19) = -2.078, p = 0.05$. That is, people switched between playing the Balloons game and watching or physically interacting with the robots less often when they were using the RobotLink app, presumably because they were using the app to observe whether the robots need attention from within the game-playing zone.

5.3.3 Robot Interactions

Button presses: There was a weak main effect of having access to the RobotLink app on the number of times participants pressed buttons on the robots [$F(1, 16) = 3.70, p = 0.07$]. There was no main effect of the experiment condition [$F(3, 16) = 1.79, p = 0.18$] and there was no significant interaction between the app and the experiment condition [$F(3, 16) = 1.04, p = 0.4$] on the number of times participants pressed buttons on the robots. A post-hoc two-tailed t-test showed (weak significance) that the number of times participants pressed buttons on the robots was fewer during the run in which they had access to the RobotLink app ($M = 10.85, SD = 9.6$) compared to the run in which it was disabled ($M = 15.7, SD = 6.56$); $t(19) = -1.92, p = 0.07$ (see Figure 5.22). Simply put, people spent less time using the robots' physical interfaces when they also had access to the RobotLink app. Two potential explanations for this are that people were using the RobotLink app controls instead of the physical controls, and that they potentially had a better understanding of why a robot might not be responding to their actions.

No one attempted to hold down the clean button to reset the robot that needed help in the first run for longer than 5 seconds without the RobotLink app (see Figure 5.23). People who did have the app in their first run appeared to try to carry over their experiences and apply them during their second run, as is evident

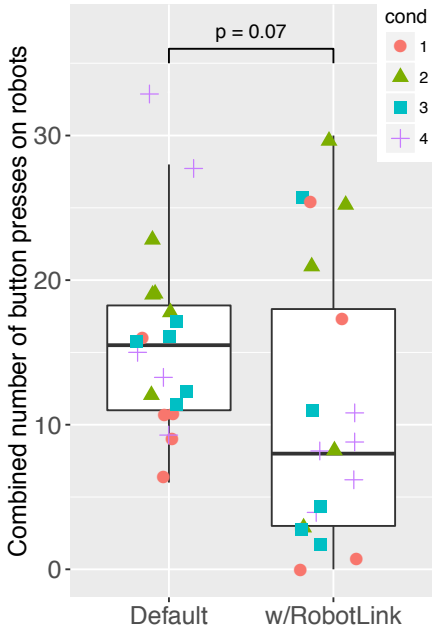


Figure 5.22: Button Presses

by the steadily increasing number of times people held down the clean button for over 5 seconds in the second run without the app. This same group of people may also have either been uncertain about which robot they needed to reset, or else thought that perhaps the same technique would work on multiple robots since they also appear to have tried to reset the "dustbin" robot more than in any other situation. Everyone who had the app in their second run successfully reset the robot that needed help, and very few attempts were made at trying to reset a robot other than the one which requested help.

Time spent resetting robot: The amount of time participants spent with the robot that needed to be reset was compared with the other two robots (see Figure 5.24). Time with robots was calculated by coding each interaction's start and stop times for every robot up until the time the robot that needed help was reset. Timing started whenever the participant knelt or leaned over a robot while

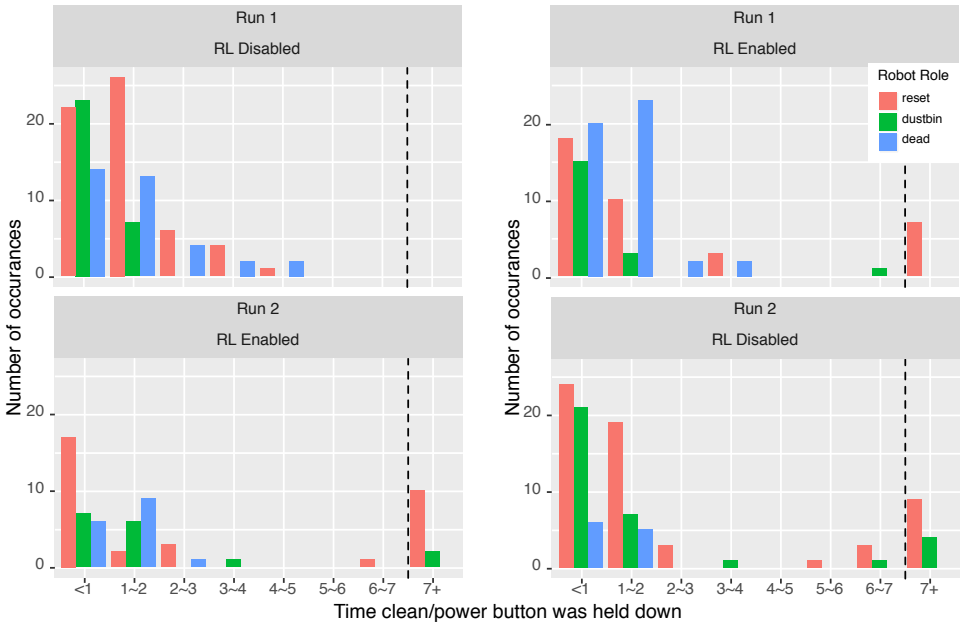


Figure 5.23: Clean button click/hold times (in seconds)

reaching towards, touching, or looking at the robot or the phone. Timing was stopped whenever the user stood up, moved their hand away from the robot, or looked away from the robot (except for when they were looking at the phone). Additionally, time was stopped if the robot moved past the visual barrier between the game-playing zone and the cleaning area (which was established as being beyond the reach of participants) or if the robot was successfully reset. Time spent working with the dustbins was excluded, with time stopping when the dustbin was removed and re-starting once it was replaced. Coded data was transformed such that discrete time segments of each run, divided into individual seconds, were designated with which of the 5 robots the participant was interacting with, or "none". Time segments that all coders unanimously marked as "none" were discarded since the focus of the coding was on time spent interacting with the robots. Inter-rater reliability of two coders (one experimenter and one researcher not involved with data collection, both included on the IRB protocol) was computed using Cohen's

Kappa and showed significant agreement ($\kappa = 0.87, \alpha = 0.05$).

A pairwise comparison using a two-tailed paired t-test on the time spent with robots during the run in which people did not have access to the RobotLink app showed a significant difference ($p = 0.03$) between the time spent with the robot that needed to be reset ($M = 28, SD = 27$) and with the robot that was playing dead ($M = 14.8, SD = 19$). The difference between the robot which needed its dustbin emptied ($M = 24.3, SD = 20$) and the robot that needed to be reset was not significant ($p = 0.6$), nor was there a significant difference between the dustbin robot and the robot playing dead ($p = 0.1$). On the other hand, a pairwise comparison using a two-tailed paired t-test of the time spent with robots during the run in which people did have access to the RobotLink app showed a significant difference ($p = 0.008$) between the time spent with the robot which needed to be reset ($M = 21.5, SD = 22$) and the robot that was playing dead ($M = 4.6, SD = 10.5$). There was also a significant difference between the robot which needed to be reset and the robot that needed its dustbin emptied ($M = 6, SD = 13.5, p = 0.02$), but not between the robot which needed its dustbin emptied and the robot that needed to be reset ($p = 0.5$). The amount of time spent with the robot that needed to be reset, combined with the distinct lack of time spent with the other two robots prior to the robot being reset, during runs with the RobotLink app indicates that participants likely understood which robot they were supposed to be resetting.

RobotLink App Use: Half of the participants (10/20) viewed all three robots' status pages using the RobotLink app. The robot that needed to be reset used a *push-style* notification which caused a popup message to appear on the smartphone screen, interrupting what the user was doing. As a result, each of the participants ended up viewing the page of robot that need to be reset at least once. The robot

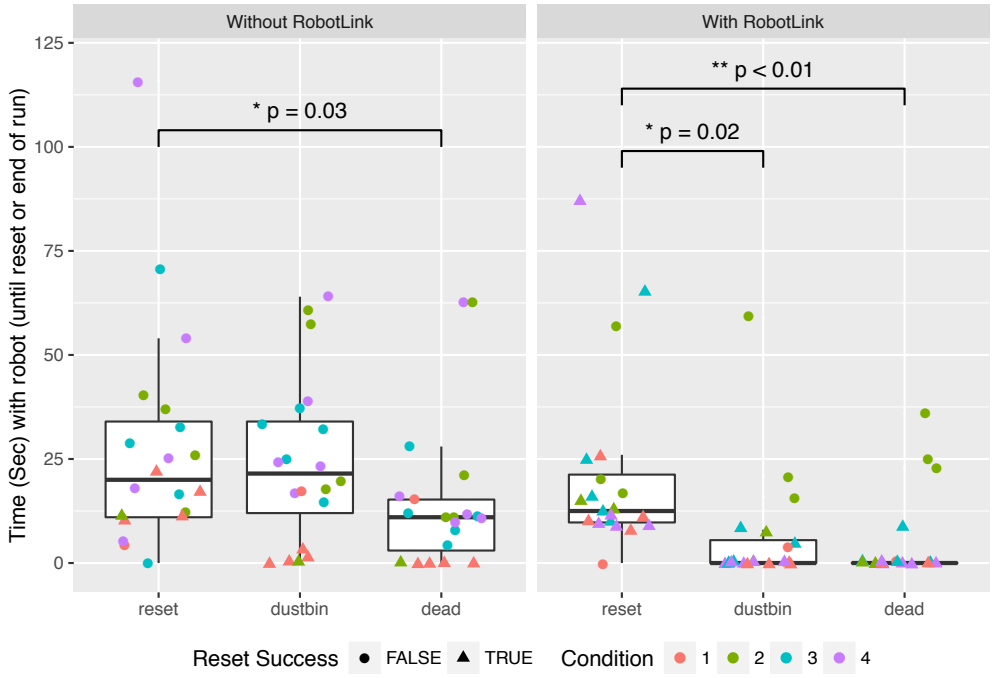


Figure 5.24: Time physically spent with robots (up until reset)

that needed its dustbin to be emptied was viewed by 12 of the 20 participants, and the robot that played dead was viewed by 10 of the 20 participants. There were no significant differences in the time spent viewing robot's status pages between conditions.

Time viewing status pages: Using the data from the 10 participants who viewed all three robots' status pages, a pairwise comparison using a two-tailed paired t-test showed significant differences in the amount of time participants spent on those pages inside the RobotLink app depending on whether the robot being viewed needed to be reset ($M = 88.7, SD = 43.7$), needed to have its dustbin emptied ($M = 45.3, SD = 17.1$), or was disabled ($M = 9, SD = 9$) (see Figure 5.25). Participants spent significantly more time looking at the page of the robot that needed to be reset than either the robot that needed its dustbin emp-

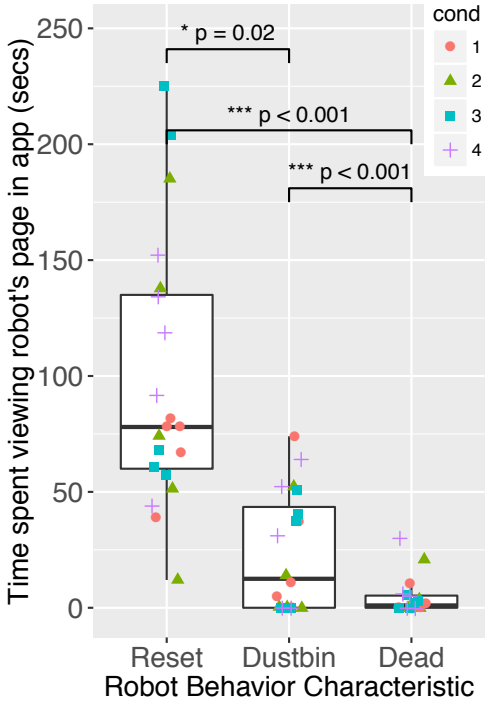


Figure 5.25: Time spent in RobotLink app

tied ($p = 0.02$) or the robot that had been disabled ($p < 0.001$). They also spent significantly more time looking at the page of the robot whose dustbin needed to be emptied than that of the robot which had been disabled ($p < 0.001$). In other words, the amount of time people spent using the RobotLink app to view information about the different robots was correlated with the appropriate amount of attention needed to get and keep each robot working.

5.3.4 Balloons Game

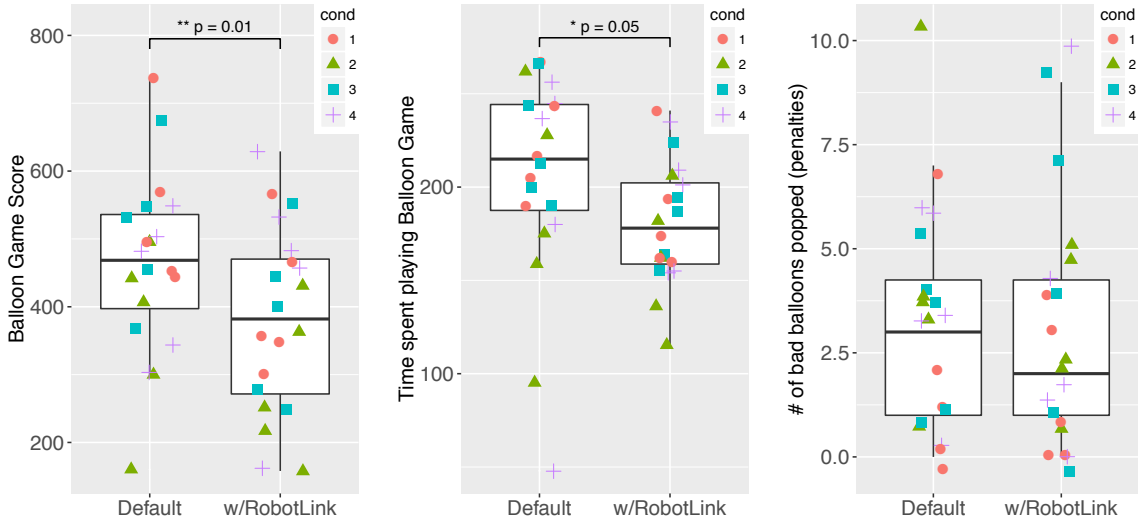
Score: There was a significant main effect of the RobotLink app on the final score of the Balloons game [$F(1, 16) = 7.567, p = 0.01$]. There was no main effect of the experiment condition [$F(3, 16) = 1.99, p = 0.15$] and there was no interaction between the app and the experiment condition [$F(3, 16) = 1.41, p = 0.27$]. A post-hoc two-tailed t-test showed that participants scored significantly fewer points

during the run in which they had access to the RobotLink app ($M = 382, SD = 137$) compared to the run in which it was disabled ($M = 463, SD = 131$); $t(19) = -2.66, p = 0.01$ (see Figure 5.26a).

Time playing: There was a significant main effect of the RobotLink app on the amount of time participants spent playing the Balloons game [$F(1, 16) = 4.52, p = 0.05$]. There was no main effect of the experiment condition [$F(3, 16) = 0.77, p = 0.52$] and there was no interaction between the app and the experiment condition [$F(3, 16) = 0.49, p = 0.69$]. A post-hoc two-tailed t-test showed that participants spent significantly less time playing the Balloons game during the run in which they had access to the RobotLink app ($M = 180, SD = 33$), compared to the run in which it was disabled ($M = 206, SD = 56$); $t(19) = -2.21, p = 0.04$ (see Figure 5.26b).

Penalties and skill: There was no main effect of the RobotLink app [$F(1, 16) = 0.13, p = 0.7$] or the experiment condition [$F(3, 16) = 0.59, p = 0.6$] on the number of "bad" (penalty) balloons that participants popped in the game (see Figure 5.26c), nor was there an interaction between the app and the experiment condition [$F(3, 16) = 0.9, p = 0.4$]. Furthermore, there was also no main effect of either the RobotLink app [$F(1, 16) = 2.11, p = 0.16$] or the experiment condition [$F(3, 16) = 1.04, p = 0.4$] on the rate at which participants scored points while playing the game. This means that people's skill level in the Balloons game did not change much, which helps explain the previous results; when people were using the RobotLink app they spent less time playing the Balloons game, and that resulted in people getting lower scores in the game.

A decrease in the number of context switches, such as we observed in Section 5.3.2, has the potential to make people more efficient during the time they



(a) Score (b) Playing time (c) Penalties

Figure 5.26: Balloons Game

spend performing a task. There was no significant difference in the rate at which participants accumulated points in the Balloons game (normalized for time spent playing the game) between runs in which participants used the RobotLink app ($M = 2, SD = 0.5$) and runs without it ($M = 2.42, SD = 1.16$); $t(19) = 1.5504, p = 0.13$. This is most likely due to the relatively simple nature of the game and the fact that it did require participants to keep track of information or required any skill.

5.3.5 Experiment Scores

Participant compensation was calculated by multiplying the Balloons game score by the percentage of beads collected by the robots, minus 100 points for robots not on their chargers, and dividing by 100 (rounding up). Unfortunately, while this formula properly motivated the participants' behaviors, in practice the calculation did not accurately represent the degree to which people achieved those behaviors.

One reason for this was that the percentage of beads collected by the robots was

not proportional to the number of robots the participant got working or the time the robots spent cleaning. Robots would occasionally get stuck in a fixed "cycle" rather than covering new area, and a number of participants spilled a non-trivial amount of beads back onto the floor while emptying the dustbins.

Another issue was the number of robots on their docks at the end of each run ended up being an unreliable metric. The time it took robots to return to their docks varied widely, but not as a function of the distance they had to travel. This made it extremely difficult for participants to predict the necessary amount of time robots needed to complete the docking process (even after being given a recommendation of 30 to 45 seconds by the experimenters.)

For analysis purposes, rather than using the calculated compensation, we measure the participant's success at the experiment as a separate score which emphasizes the time (the two working) robots spent running and the Balloons game score:

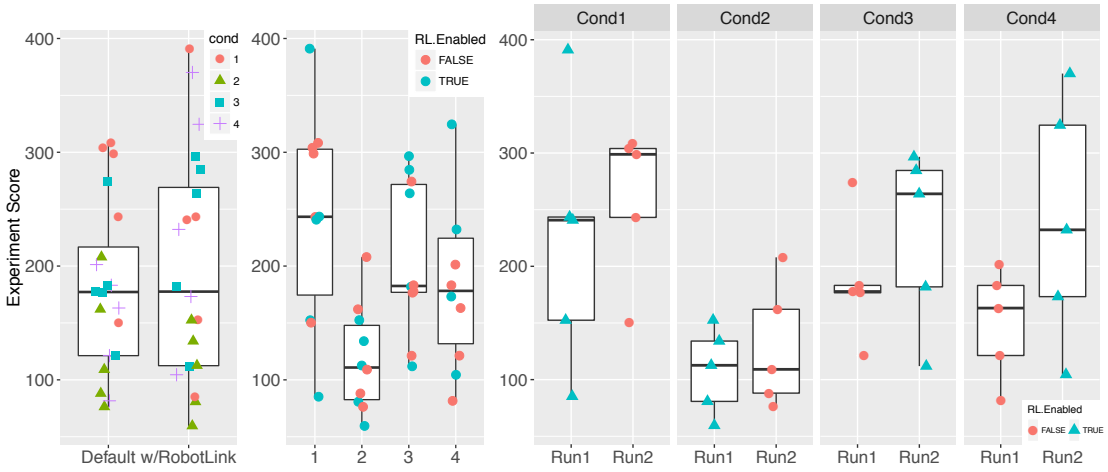
$$Score_{\text{Expmt}} = Score_{\text{Balloons}} \times \frac{\sum_{\text{Robots}} Time_{\text{Cleaning}}}{Time_{\text{Expmt}} \times 2} \tag{5.1}$$

There was no main effect of the RobotLink app on the experiment score [$F(1, 16) = 1.16, p = 0.3$] (see Figure 5.27a). However, there was a significant main effect of the experiment condition on the experiment score [$F(3, 16) = 3.18, p = 0.05$], and there was a weak interaction between the app and experiment [$F(3, 16) = 3.06, p = 0.06$]. A post-hoc pairwise comparison using paired t-tests showed significant differences between Condition 2 ($M = 196, SD = 75$) and Conditions 1 ($M = 184, SD = 92, p = 0.03$) and 3 ($M = 183, SD = 78, p = 0.04$).

This means that the RobotLink app by itself did not make much of a difference in people's overall performance. Instead, we found that the experiment condition

people were placed in, which should not have made a difference, did have an influence on their experiment score. In particular, participants from Condition 2 seemed unable to score as high as people in the three other condition groups (see Figure 5.27b). While people typically got higher scores during their second run, this was not as pronounced in Condition 2 (see Figure 5.27c).

We have been unable to produce a suitable explanation for why participants in condition 2 did not perform as well as those in the other three conditions. The low experiment scores were a combination of both low Balloon game scores (although the differences were not significant) and low combined time robots spent cleaning. The later was the result of just a single person (out of five) in Condition 2 successfully using more than one robot, compared to 4 out of 5 in Condition 1, 5 out of 5 in Condition 3, and 4 out of 5 in Condition 4. There were no significant differences between Condition 2 and the other conditions with respect to time spent outside the game-playing zone, RobotLink app usage, time spent playing the Balloons game, penalties incurred in the Balloons game, or time spent physically interacting with the robots.



(a) RobotLink (b) Conditions (c) Runs, by condition

Figure 5.27: Experiment Score

5.3.6 Post-Run Questionnaires

Following each run, participants were asked to fill out a questionnaire (see Appendix A.2.2) asking how they felt about their experience working with the robots, their perception of what help (if any) they needed to provide to robots, and their workload.

5.3.6.1 Understanding Requests for Help

In each of the two runs, the three robots each engaged in one of three distinct behaviors. One robot immediately required being "reset" before it could begin cleaning, but once this had been completed could continue running without needing any other help. A second robot was immediately available to begin cleaning upon request, but thereafter needed to periodically have its dustbin emptied before it could continue cleaning. The last robot played dead, and simply refused to work for the entire duration of the run. During both runs, the two robots that needed help would emit visual (flashing lights) and auditory indicators (beeping sounds) to signal there was a problem until the issue was resolved. Following each run, participants were asked *"What kind of help did the robot(s) require or request? Select all that apply."* (see Figure 5.28a). With a single exception, participants' responses were limited to the two actions which they actually needed to take. More people correctly identified the two solicited actions during the second run (24) than during the first run (19).

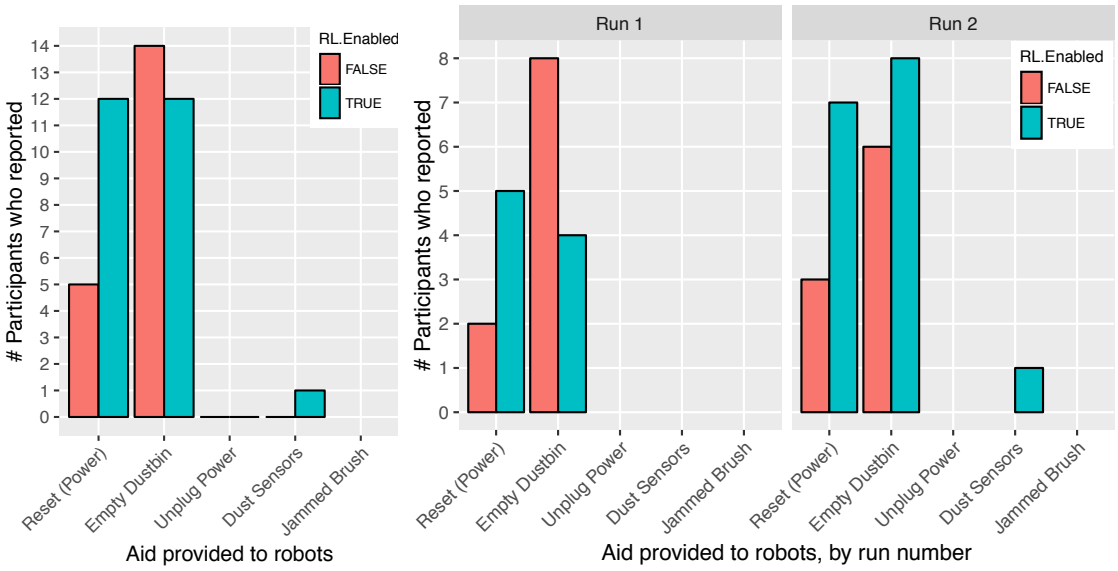
Reset: The number of participants who understood they needed to reset one of the robots was significantly higher ($p = 0.04$ using McNemar's test) during runs in which participants had access to the RobotLink app (12/20) than during runs in which it was disabled (5/12). The run number did not significantly effect

participants' understanding of whether or not a robot needed to be reset ($p = 0.5$). However, the order in which the phone was used did seem to have an effect; 5/10 participants who had access to the RobotLink app during the first run understood they needed to reset one of the robots, compared to 7/10 who had the app during the second run. In comparison, without the app only 2/10 people during the first run and 3/10 from the second run understood that one of the robots needed to be reset.

Emptying Dustbin: There were no significant differences between the runs in which the RobotLink app was enabled and those where it was disabled ($p = 0.68$ using McNemar's test), nor was there a significant difference between run numbers ($p = 0.68$) with respect to the number of people who reported needing to empty a robot's dustbin. Nonetheless, twice as many of the participants who had access to the RobotLink app during the first run (4/10) understood they needed to empty the dustbin of one of the robots during the second run (8/10). During the first run without the app, 8/10 people correctly understood that emptying the robot's dustbin was helpful to the robot. However, of the ten people who had the app in the first run, only six reported needing to empty one of the robots' dustbins. It is worth emphasizing that these results reflect participants' perception of their experiences, rather than differences in the actual scenario they encountered (which was the same across all conditions and runs).

5.3.6.2 Working with the Robots

Confidence: A two-tailed paired t-test showed that people reported having significantly more confidence in their understanding of the robots' behaviors during runs in which they had the RobotLink app ($M = 4.65, SD = 1.57$) compared to



(a) Total responses

(b) Responses by run number

Figure 5.28: Help participants reported giving to robots

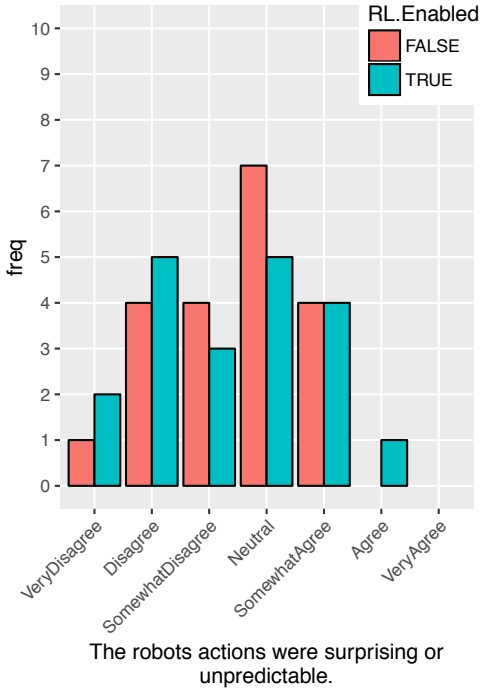
runs in which the app was disabled ($M = 3.7, SD = 1.42$); $t(19) = 3.13, p = 0.005$ (see Figure 5.29b). Without the RobotLink app, 11/20 people described their confidence as "Moderately Low" to "Very Low", 5/20 people reported their confidence as acceptable, and 4/20 ranked their confidence "Moderately High" to "High". During runs where participants had access to the RobotLink app, only 6/20 people described their confidence as "Moderately Low" to "Very Low" and 2/20 people reported their confidence as acceptable, while 12/20 ranked their confidence "Moderately High" to "High". In other words, the app made people feel more confident that they understood what assistance the robots needed.

Robot predictability: Participants generally felt that the robots were predictable (see Figure 5.29a) regardless of whether or not they had access to the RobotLink app (there was no significant difference found using a two-tailed paired t-test; $t(19) = -0.31, p = 0.7$). Sixteen out of 20 people during the runs without

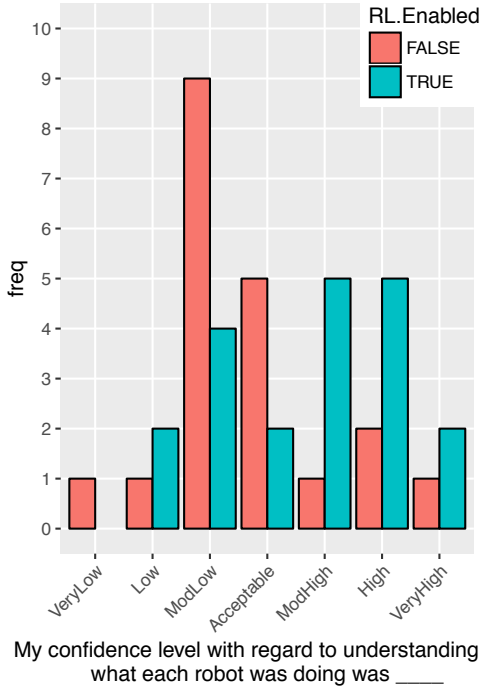
the RobotLink app and 15/20 people during runs with the app either disagreed or were neutral when asked if they felt the robots actions were surprising or unpredictable. Only 4/20 felt the robots were "somewhat" unpredictable in either app condition, and a single person felt that during one of their runs (with the app) that the robots were behaving unpredictably.

Satisfaction with robots: People's satisfaction with the robots was higher (weak significance) during the runs in which the RobotLink app was enabled ($M = 4.6, SD = 1.19$) than during runs in which it was disabled ($M = 4.05, SD = 1.23$) according to a two-tailed paired t-test; $t(19) = 1.93, p = 0.07$ (see Figure 5.29c). Ten out of 20 people ranked their satisfaction as "Moderately High" to "Very High" during the run in which the app was enabled, compared to 5/10 during the run in which it was not.

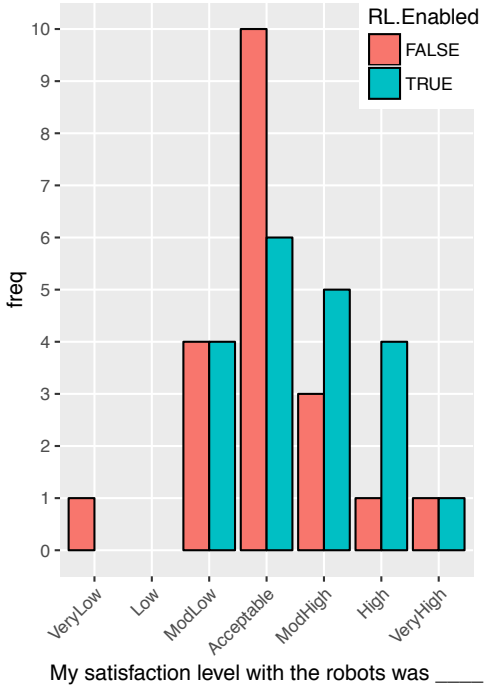
Determining what to do: A two-tailed paired t-test showed that participants found it significantly easier to determine what they needed to do to make each robot work during the run in which they had the RobotLink app ($M = 5, SD = 1.26$), compared to the run in which the app was disabled ($M = 3.7, SD = 1.75$); $t(19) = 3.21, p = 0.004$ (see Figure 5.29d). Fifteen out of 20 people reported it was "Moderately Easy" to "Very Easy" to determine what they needed to do during the runs that the app was enabled, compared to only 7/20 people during the runs where it was disabled. On the other hand, 13/20 people reported it being "Moderately Difficult" to "Very Difficult" during runs in which the app was disabled, compared to only 4/20 people during the runs where it was enabled.



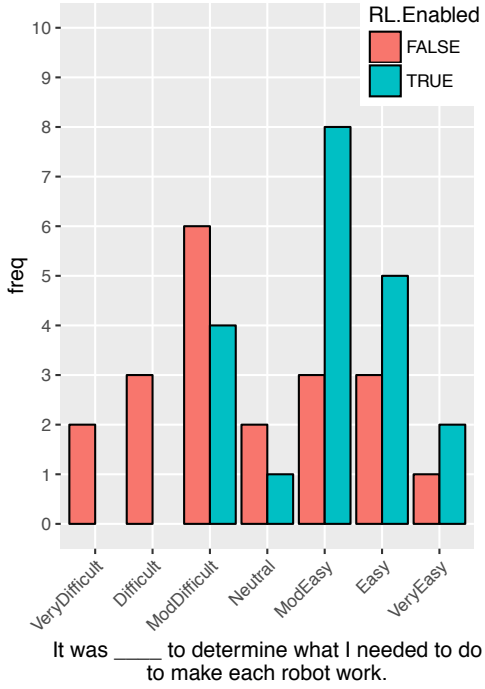
(a) Robot predictablity



(b) Confidence



(c) Satisfaction



(d) Obtaining Information

Figure 5.29: Participant robot reviews, by RobotLink app usage

5.3.6.3 Workload (NASA TLX)

Participants were issued NASA's Task Load Index (TLX) Questionnaire [Hart and Staveland, 1988] after each run. Six 2x2 mixed-groups factorial ANOVA were performed to examine the effects of using the RobotLink app and experiment conditions used on participants' mental and physical workload, their perceived performance and success, and how finally how rushed and discouraged they felt.

Mental workload: There was a weak significant main effect of having access to the RobotLink app on the mental workload reported by participants [$F(1, 16) = 4.26, p = 0.055$] (see Figure 5.30a). Additionally, there was a significant main effect of the experiment condition on mental workload [$F(3, 16) = 3.3, p = 0.04$], and a weakly significant interaction between the RobotLink app and the experiment condition on mental workload [$F(3, 16) = 3.02, p = 0.06$]. A post-hoc two-tailed paired t-test showed (with weak significance) that participants tended to have a lower mental workload during the run in which they had the RobotLink app ($M = 3.7, SD = 1.13$) than the run without it ($M = 4.1, SD = 1.33$); $t(19) = -1.8, p = 0.088$. A pairwise two-tailed t-test with Holm correction that was used to compare differences in mental workload between experiment conditions found significant differences between condition 1 ($M = 4.8, SD = 1.14$) and conditions 3 ($M = 3.4, SD = 0.97, p = 0.048$) and 4 ($M = 3.1, SD = 0.074, p = 0.045$), but not between any of the other experiment conditions.

These findings suggest that using the RobotLink app lowered people's mental workload, but that peoples perception of their workload was influenced by their prior experience. We found that people who were placed in Conditions 3 and 4 for the experiment had lower mental workloads than people in Condition 1. This is interesting because people used the RobotLink app during their second run for

both Conditions 3 and 4, while people in Condition 1 used the app in the first run (the same was true of Condition 2, but the difference was not as pronounced as in Condition 1.)

Physical workload: There was a significant main effect of having access to the RobotLink app on the physical workload reported by participants [$F(1, 16) = 4.65, p = 0.046$] (see Figure 5.30b); however there was no main effect of the experiment condition [$F(3, 16) = 1.80, p = 0.18$] on physical workload, nor was there a significant interaction between the variables [$F(3, 16) = 1.06, p = 0.39$]. A post-hoc two-tailed paired t-test showed that participants tended to have a lower physical workload during the runs in which they had the RobotLink app ($M = 2.15, SD = 0.81$) compared to the runs in which the app was disabled ($M = 2.7, SD = 1.34$); $t(19) = -2.15, p = 0.04$. This result is intuitive; the RobotLink app allowed people to substitute needing to walk between different locations in the room and reaching down to access robots' controls with a click of a button.

Feeling rushed: There was a significant main effect of having access to the RobotLink app on how rushed participants reported feeling [$F(1, 16) = 4.26, p = 0.05$] (see Figure 5.30d); however there was no main effect of the experiment condition [$F(3, 16) = 2.12, p = 0.14$] on participants feeling rushed, nor was there a significant interaction between the variables [$F(3, 16) = 2.02, p = 0.15$]. A post-hoc two-tailed paired t-test showed that participants tended to feel more rushed during the runs in which they had the RobotLink app ($M = 4.45, SD = 1.19$) compared to the runs in which the app was disabled ($M = 4, SD = 1.45$); $t(19) = -1.9, p = 0.07$. A trend in Figure 5.30d seems to indicate that participants who had the app in the first run tended to feel more rushed during one or both of the

runs. Although the trend was not statistically significant, it is not difficult to imagine that after being introduced to the app people felt a little more overwhelmed by this additional aspect that was now competing with the other tasks they needed to balance their attention between, especially when this occurred during the first run.

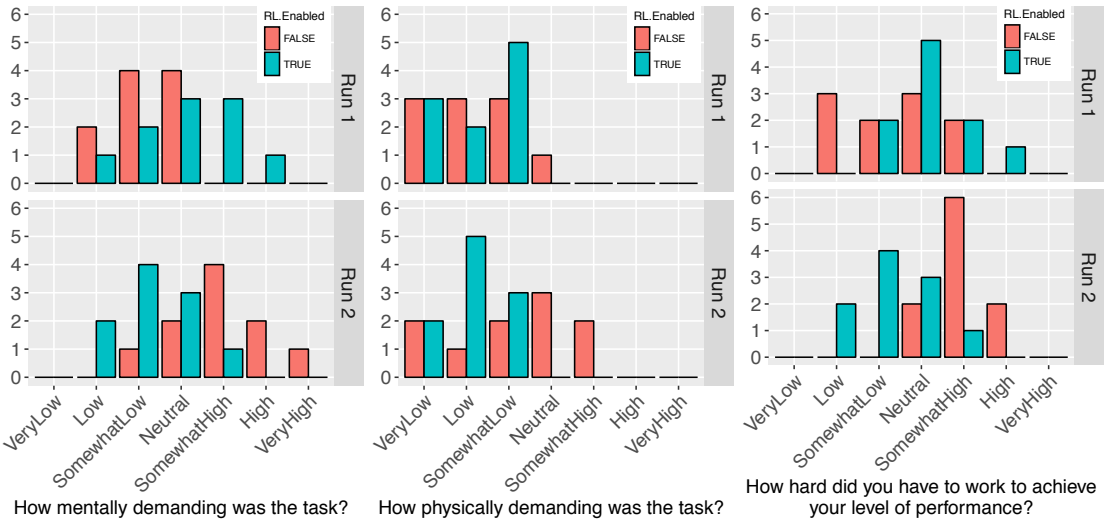
Performance: There was a weakly significant main effect of having access to the RobotLink app on how hard participants reported having to work to achieve their level of performance [$F(1, 16) = 3.6, p = 0.07$] (see Figure 5.30c). There was a significant main effect of the experimental condition on how hard participants reported having to work [$F(3, 16) = 4.77, p = 0.01$], as well as a significant interaction between the two variables [$F(3, 16) = 3.24, p = 0.05$]. A two-tailed paired t-test failed to show a significant difference between runs in which participants had access to the app ($M = 3.75, SD = 1.02$) compared to runs in which they did not ($M = 4.2, SD = 1.24$); $t(19) = -1.63, p = 0.11$. A pairwise two-tailed t-test with Holm correction that was used to compare differences in how hard participants worked to achieve their level of performance between experiment conditions found significant differences between Condition 1 ($M = 4.7, SD = 0.82$) and Condition 3 ($M = 3.2, SD = 1.03, p = 0.002$), and weak significance between Conditions 3 and 2 ($M = 4.5, SD = 1.03, p = 0.066$), but not between any of the other conditions.

The differences between experiment conditions seems to be correlated with the order in which the RobotLink app was used during runs, an explanation which is consistent with the above findings and similar to our observations regarding mental workload. Figure 5.30c shows what appears to be a reversal of opinions between the first and second runs based on whether or not the app was being used. People who used the app in their second run felt that it took less effort to achieve their level of performance, while those who used the app in their first run felt that it

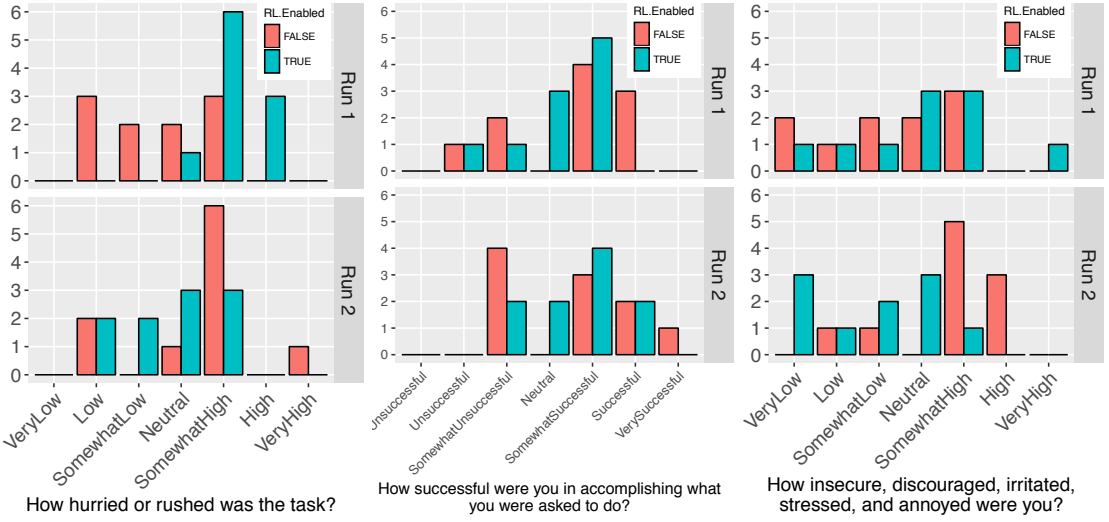
took somewhat more effort to achieve their performance level than those who did not have the app.

Feeling discouraged: There was a significant main effect of having access to the RobotLink app on how discouraged participants reported feeling [$F(1, 16) = 5.04, p = 0.04$]; however there was no main effect of the experiment condition [$F(3, 16) = 1.54, p = 0.24$], nor was there a significant interaction between the variables [$F(3, 16) = 0.34, p = 0.79$] (see Figure 5.30f). A post-hoc two-tailed paired t-test showed that participants reported feeling less discouraged during the run in which they had the RobotLink app ($M = 3.4, SD = 1.67$) than in the run without it ($M = 4.05, SD = 1.62$); $t(19) = -2.37, p = 0.03$. The fact that people felt less discouraged when using the RobotLink app is consistent with our findings that people felt more confident they understood what the robots were doing and also found it easier to figure out what they needed to do to make robots work while using the app (see Section 5.3.6.2).

Success: There were no significant main effects of having access to the RobotLink app [$F(1, 16) = 0.42, p = 0.5$] or the experiment condition [$F(3, 16) = 0.4, p = 0.75$] on how successful people felt (see Figure 5.30e). There was also no significant interaction between the two variables [$F(3, 16) = 0.49, p = 0.7$]. People generally tended to believe they had been successful, although a few people did feel otherwise. Our results indicate that more people felt unsuccessful to some degree in cases where they were not using the RobotLink app (7/20) compared to the cases where it was being used (4/20), however this difference was not enough to be statistically significant.



(a) Mental Load (b) Physical Load (c) Performance



(d) Rushed (e) Success (f) Discouraged

Figure 5.30: NASA TLX results, by run

5.3.6.4 RobotLink App information

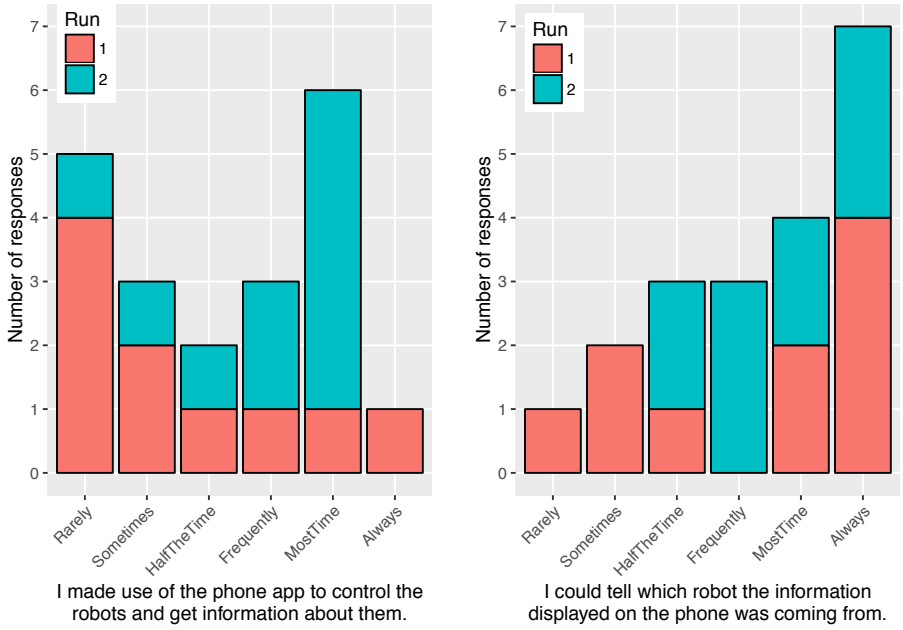
After the run in which the RobotLink app was enabled, participants were asked questions related to their usage of the app and the information it provided. There was a weak trend toward participants who used the RobotLink app in the second run feeling like they made more use of it than those who used it during their first run ($M_{Run1} = 3.6, M_{Run2} = 4.9, t(17) = -1.75, p = 0.09$, using a two-tailed t-test). Seven of the 10 participants who had the RobotLink during the second run felt they used the app more than half the time, while 6 of the 10 participants who had the app during the first run felt they used it less than half the time (see Figure 5.31a).

The majority of participants (14/20) felt they understood which robot the information in the app was referring to more than half the time, while only 3/20 felt they "rarely" or "sometimes" understood. A two-tailed t-test did not report a significant difference ($t(14.6)$=-0.5, p=0.6) between run ordering conditions (see Figure 5.31b).

5.3.7 Post-Experiment Questionnaires

After completing the second run, participants were asked to fill out a final questionnaire (see Appendix A.2.3). When asked where they believed the information in the RobotLink app came from, about half of the participants (11/20) believed that it was coming directly from the robots, 14/20 believed that it came from a remote system controlling the robots, and 3/20 believed it came from another person. There were no significant differences between run ordering conditions (see Figure 5.32).

Nearly all of the participants (18/20) reported that learning how to use the RobotLink app was easy (see Figure 5.33a). Most participants also reported that



(a) Usage (b) Which robot?

Figure 5.31: RobotLink app usage

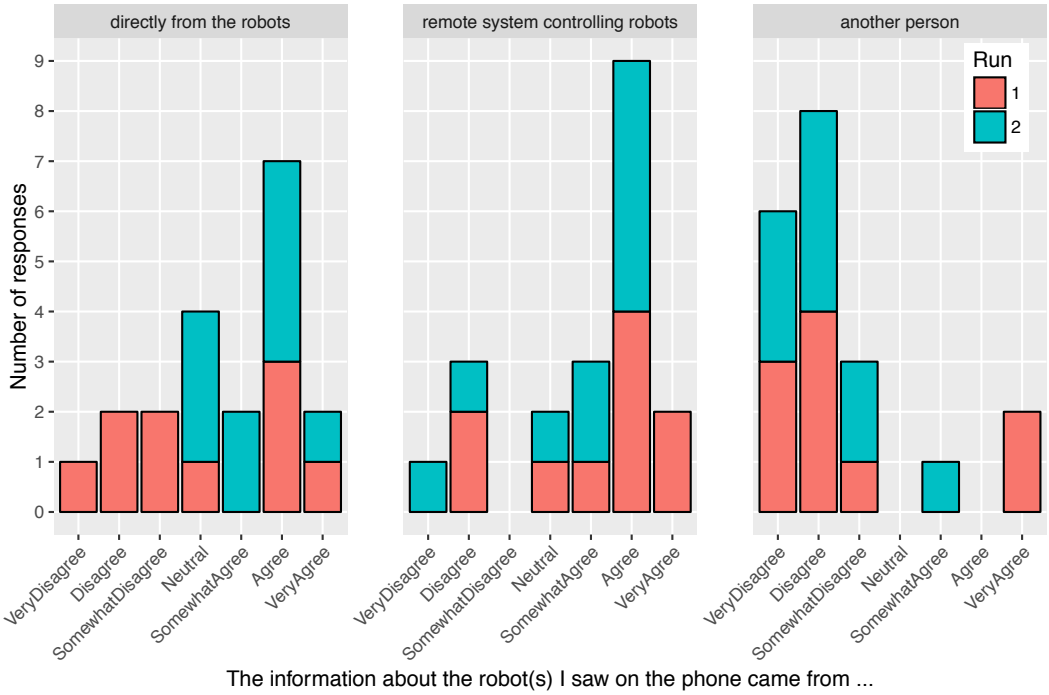
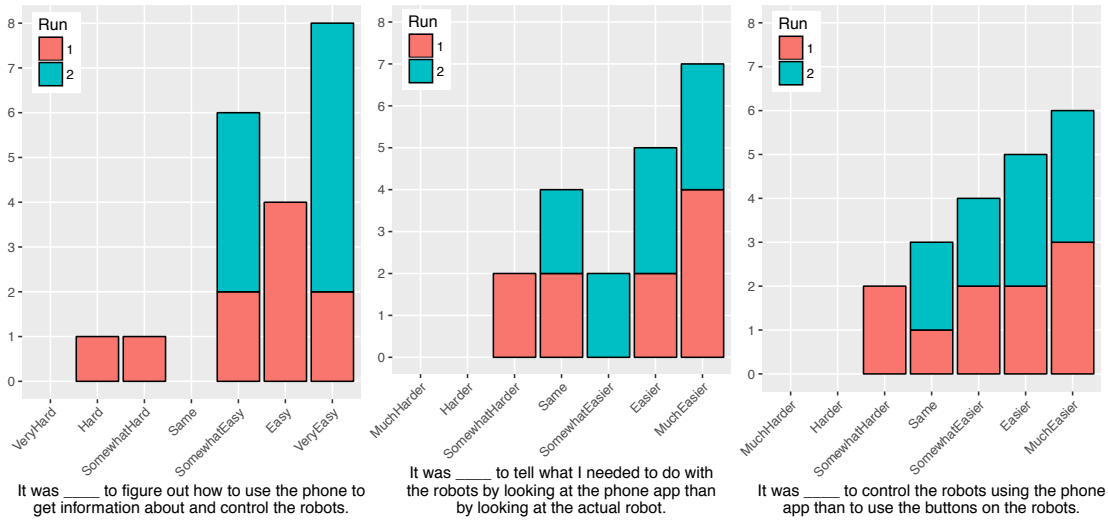


Figure 5.32: Perceived RobotLink app information source

it was easier to figure out what they needed to do with the robots (14/20) and to control them (15/20) by using the RobotLink app rather than by looking at the robots themselves (see Figures 5.33b and 5.33c).

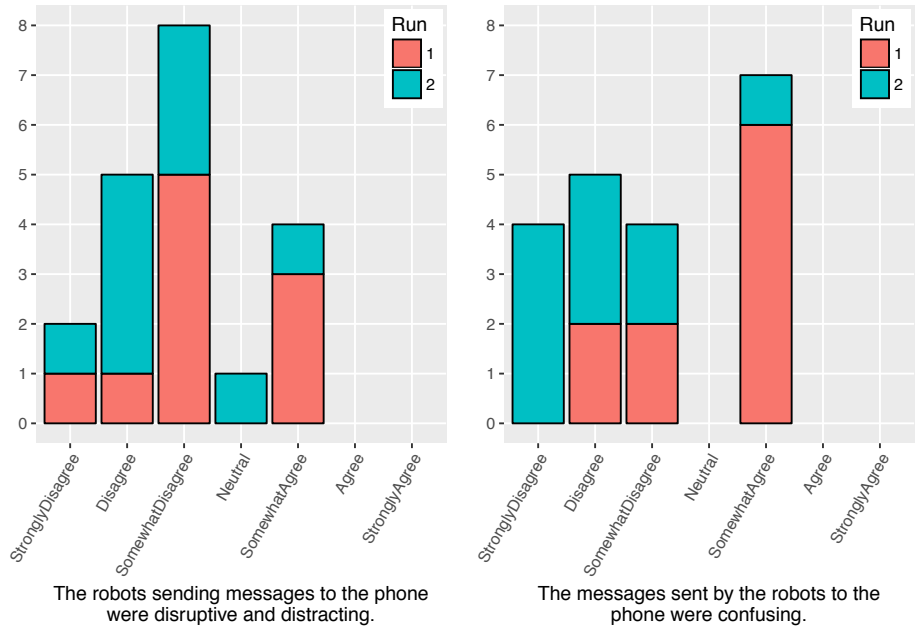


(a) Learnability (b) Information (c) Control

Figure 5.33: RobotLink app characteristics

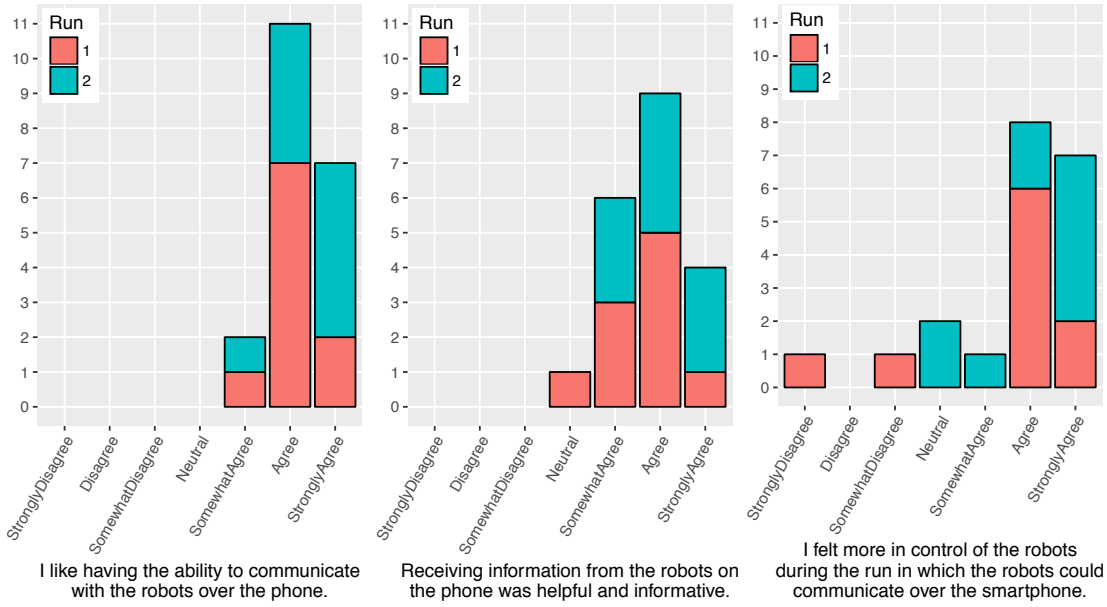
The majority of participants (15/20) felt that the RobotLink app was not disruptive or distracting (see Figure 5.34a). However, most of those who did think it was disruptive used the phone during their first run. There was significant disagreement over whether or not information in the app was confusing (see Figure 5.34b), with 6/10 participants who used it during the first run rating it as "somewhat confusing", while 9/10 participants who used it during the second run disagreed ($M_{Run1} = 4, M_{Run2} = 2.1, t(18) = 3.24, p = 0.004$ using a two-tailed t-test).

All 20 participants said they preferred having access to the RobotLink app (see Figure 5.35a). Nineteen out of the 20 participants thought the information they received from the app was helpful and informative, and 16/20 felt more in control of the robots when they had access to the app (see Figures 5.35b and 5.35c). There were no significant differences between run ordering conditions ($p = 0.3, p = 0.3$, and $p = 0.4$, respectively, using two-tailed t-tests).



(a) The app was distracting (b) The app was confusing

Figure 5.34: RobotLink app negative traits



(a) Preferred the app (b) App was helpful (c) Feeling of control

Figure 5.35: RobotLink app preferences

In summary, we have made the following observations based on our results:

- People were able to get more robots working and keep robots cleaning longer with the RobotLink app.
- People context switched between playing the Balloons game and working with the robots less often with the RobotLink app.
- There were significant differences in the amount of physical interactions people had with robots based on whether or not they were using the RobotLink app.
- People spent less time playing the Balloons game when using the RobotLink app, which resulted in lower scores.
- Participants' experiment scores were not significantly different between runs with the RobotLink app and runs without it.
- Participants felt it was easy to figure out how to use the RobotLink app, thought it was helpful, and preferred having access to it.
- Most people believed they could tell where information in the RobotLink app was coming from.
- People were more confident they understood what robots were doing while using the RobotLink app.
- Participants modified their behavior when using the RobotLink app to better target correct robots.
- Half the people were able to use the RobotLink app to view information about 2 or more robots.

5.4 Discussion

The results support H5.1, that participants were able to determine which robot they were communicating with using the RobotLink app. A majority of people reported that they thought they could tell which robot the information in the app came from (70%) and that it was easier to figure out what they needed to do to make the robots work (70%) with the app than by looking at the robots. Six out of the ten participants who had the app during their first run were able to "reset" the robot that needed help, and five out of ten were able to transfer that knowledge, by successfully identifying and resetting a different robot that needed help in their second run without using the app.

In contrast, no one was able to reset the robot that needed help during the first run without using the RobotLink app. However, *all of those same people* (10/10) were able to successfully reset another robot which needed help when given the RobotLink app in their second run.

The results partially support H5.2. Participants were able to retrieve information about the robots with which they were working to identify solutions to problems. All of the participants used the RobotLink app to view the robot that used a *push-style* interaction to cause a popup message to appear on the phone, and half of the participants viewed all three robots using the app. There are a few possible reasons why more of participants did not view all three robots.

First, while the basic use and controls of the Android phone were demonstrated to participants, including how to switch between apps, they were not provided with any training on how to use the RobotLink app or instructions concerning that app's capabilities or use cases. The popup message appeared shortly after the run began, making it one of the first experiences participants had with using the RobotLink app. Therefore, they may have believed that if any of the other robots

had information for them they would receive it in a popup message as well, so it never occurred to them to try looking at the app for information they were not told about.

Another related possibility is that some participants may have simply not been interested in viewing the information of some robots, such as in the cases of two participants who viewed 2 of the 3 robots (in both cases they choose not to view the robot that was playing dead). The third possibility is that some participants may not have been able to figure out how to access the information about the other robots. Of the 8 participants who only viewed one robot, 5 of them reported having no prior experience using Android devices. While the app would initially open to show the list of nearby robots in the "Robot Chooser" activity, it would also remember and open to the last page the user viewed (a standard app behavior). Since all of the participants viewed the robot that needed to be reset after seeing the popup message, some of the non-Android users may have had difficulty figuring out how to navigate back to the list of robots (despite this being explained in the script read by the experimenter — see Appendix A.1).

People were able to use the app to identify solutions to problems and allocate their time more appropriately. Participants were much more likely to get 2 robots working during the run with the RobotLink app and as a result had a combined time for robots cleaning that was over a minute and a half longer (on average) than times in runs without the app. During the runs in which participants had the RobotLink app, they tended to focus most of their time and attention on the robot which needed to be reset first.

People using the app were also able to get the robots to clean for longer while also spending less time watching and physically interacting with them. We had predicted that this kind of behavior would lead to better performance since par-

ticipants would have more time to score points in the Balloons game and be able keep a higher percentage of their score. However, despite gaining an average of 30 additional seconds which could be used to play the Balloons game during the run in which the RobotLink app was enabled, people using the app actually spent significantly less time playing the Balloons game, causing their overall performance to be about the same as it was without the app. Much of the lost time was spent using the RobotLink app (see Figure 5.36). One potential explanation for this behavior is that the RobotLink app's ability to communicate with and control the robots had a strong novelty effect on participants, leading them to spend more time with it than with the Balloons game. This explanation is supported by the fact that only 3 participants had previously used a robot vacuum cleaner.

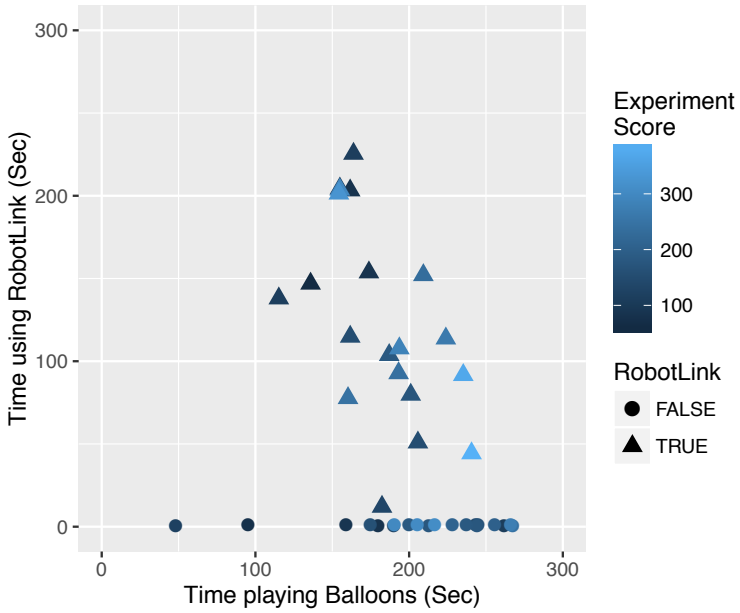


Figure 5.36: Balloons game time vs RobotLink app usage

The results of the questionnaires support H5.3. Participants liked having access to the RobotLink app, the additional information it provided, and its controls over the robots' built-in interfaces. All of the participants reported that they liked being

able to communicate with the robots through the app, and all but one (who was neutral) thought the app was helpful. The majority of participants said they felt more in control using the app, and also that they were much more confident about their understanding of what the robots were doing and what they needed to do to get them to work. According to our NASA TLX questionnaires, people felt less discouraged while using the app. Unsurprisingly, people also reported higher levels of satisfaction with using the robots during the run with the app. According to workload data from the NASA TLX questionnaires, the app also reduced people's mental and physical workloads.

5.4.1 Effects of Run Ordering

Experiment conditions were counter-balanced such that half of the participants experienced the RobotLink app in their first run while the other half used it during their second run. Nonetheless, some of our results still show evidence of the ordering effect. For example, participants who used the app in the first run were able to carry over knowledge from their first experience and to apply it during their second run as well (e.g. resetting the robot which needed help). On the other hand, the 100% increase in participants who could reset the robot in the second run with the app who had previously been unable to without it, suggests that perhaps they had a better understanding of the significance of the information the app was giving them because of their prior experience. This same group also had a higher percentage of people who understood what kinds of help they were having to provide the robots.

Another place where some ordering effects can be seen is in the NASA TLX responses, which participants answered immediately after each run. A number of participants said they found the app to be distracting (4/20) or its messages confusing (7/20), with most of the complaints coming from people who used the

app during the first run. Participants who used the app during their second run had the additional context of having also performed the task without it, while participants who used the app during their first run lacked this perspective. The results from a similar question asked at the end of the experiment support this theory; in that question, all but one participant said they thought the information from the phone was helpful.

5.5 Limitations and Future Work

There were several limitations to this study. Perhaps the most important limitation is that the participants in this task were not representative of bystanders, an important target audience for this work. Instead, the participants in this study were acting as operators (or supervisors) with the robots' goals being aligned with their own goals. The robots' primary behavior of cleaning the floors was only activated after being explicitly commanded by the participant, who only needed to observe whether or not the robot subsequently behaved as they expected it to. That said, our participants also shared several characteristics of typical bystanders: a lack of familiarity or prior experience with the robot platform and a lack of training with its user interfaces. However, given the significance of bystanders in the interface design for RobotLink, an experiment explicitly testing its use in bystander situations is merited. Ideally, we would like to test true bystander situations in which people are required to interact with a robot without previously having been informed that that the experiment would involve a robot at all, or that the robot was even part of the experiment. An alternatively, less deception oriented experiment might ask people to find and help an autonomous robot carry out a task it has been assigned without prior knowledge of exactly where the robot is, what it looks like, what it

is doing, or how to communicate with the robot.

Another important limitation in this work was the lack of explicit testing of the effectiveness of *push-style* vs *pull-style* interactions. Both interaction styles were used during the experiment, however, they were directly paired with a single type of problem, and always occurred in the same order with participants receiving a *push-style* interaction popup message shortly after the beginning of the run. Furthermore, instructions regarding how to empty the dust bins of the robots may have influenced people to think of this as a possible solution to problems without needing additional cueing, which would have eliminated the need for the *pull-style* interaction. Better testing of the difference in effectiveness between these interaction styles is warranted; for example, using a between subjects design with different types of information and the interaction style as independent variables would provide better information about how effectively people can acquire information about completely unfamiliar situations using these methods.

5.6 Conclusions

Smartphones may be a viable platform for implementing a ubiquitous interaction style which allows bystanders to communicate with autonomous robotic services in the future, which is supported by the results of this work. Our participants felt the app was easy to learn and use, despite not receiving any training and having less than 7 minutes to use it. All of the participants preferred having access to the app, and all but one said the app was was helpful. Participants were able to retrieve information about nearby robots, and could distinguish where information coming through the app had come from despite similarities in the robots appearances. While using the app, participants felt more confident they understood what robots

were doing and were more satisfied with the machines' performance. Although participants' experiment scores did not improve with the use of the app, their behavior (specifically, spending less time watching and physically interacting with the robots, and getting more robots working for longer periods of time) created the potential for improved performance. Additional experimentation is needed to better understand the differences between the *push-* and *pull-style* interaction methods and how bystanders might use the system. These results are a promising first step towards building communication between people and the autonomous robots with which we will soon be sharing our society.

Chapter 6

Design Guidelines

We believe that for autonomous robots to be fully accepted into society, they need to be capable of gracefully handling both technical and social failures. In other words, we need failure-ready robots. In this chapter, we discuss some widely used human-computer interaction (HCI) design guidelines and how they can be applied when designing interfaces for failure situations with examples from our work. We also propose four design principles specifically for creating failure-ready robots.

6.1 Applying HCI Design Guidelines to Failure Scenarios

We found that many of the major HCI design guidelines, such as those proposed by Shneiderman [2010], Norman [2013], and Nielsen [1994], were applicable in the design of the two user interactions developed for this dissertation. In this section we take a look at how these guidelines were specifically appropriate within the context of creating user interfaces for situations in which a robot is either failing or not operating as per the user's expectations.

6.1.1 Shneiderman's Eight Golden Rules

In his book, *Designing the User Interface* [Shneiderman, 2010], Shneiderman listed "eight golden rules" that are most applicable to interactive systems. We have taken some liberties in interpreting these rules in order to tune them for use in the context of our minimalistic interfaces. Our interfaces were designed more with a focus on informing the people using them and allowing them to makes small adjustments to existing situations than for providing full-featured control mechanisms.

Cater to universal usability. Shneiderman writes that designers should *"Recognize the needs of diverse users . . . Adding features for novices, such as explanations, and features for experts, such as shortcuts and faster pacing, can enrich the interface design and improve perceived system quality."* Furthermore, Shneiderman stresses that interfaces should specifically consider users of different ages, the various disabilities that people may have, and people's proficiency with technology.

The user interfaces described in this dissertation do not cater specifically to expert users. However, they are intended to be usable by as wide an audience as possible. In particular, our use of icons decouples information provided by the robot from any specific language, and their crowdsourced design leverages people's preconceived notions about the meanings of particular shapes, symbols, and colors to help them correctly characterize the state of an autonomous robot without the need for any prior explanation of the icons' significance.

In addition to embracing the original intent of this rule, we also added another dimension — universal applicability. We tested our icons on 14 different robot platforms with a wide variety of characteristics. Standardization of these kinds of icons can help to further increase people's recognition and awareness of their meanings, thereby providing a reliable mechanism through which autonomous robots could

explicitly improve nearby people's situation awareness.

Offer informative feedback. Shneiderman originally specified this as "*For every user action, there should be system feedback.*" Within the context of graphical user interfaces, actions typically refer to user input, while system feedback provides confirmation that the system understood the user and also indicates the results of the action.

Autonomous robots differ from traditional computer applications in that their system state is subject to change not only by user input, but also by a variety of other factors the user may not be aware of. Furthermore, interacting with an autonomous system is not limited to just the end users or operator of a robot; bystanders will also need to interact with these machines. For example, a pedestrian will want to confirm that a self-driving car has recognized their presence before stepping out into a crosswalk.

We have applied this concept of feedback to the general state of the autonomous system (such as if the robot is operating properly and how safe it is) and the high-level behaviors which it executes (what directive or task it is currently executing). State icons displayed on a robot should immediately reflect any changes in the characteristics of the system. For example, if a system has been running without issue but then encounters an intractable obstacle to achieving its goal which requires the assistance of a person to be resolved, the system's state icon should change from `OK` to `HELP`. Our RobotLink application was designed to allow anyone to query for basic amounts of information from nearby autonomous systems in order to better understand what they are doing and, in appropriate circumstances, provide them with some level of control over how those systems should behave. During our user study, the robot's current behavior was reflected within our app. For example,

the app might inform participants that the robot was "cleaning", "looking for its dock", or "docked" depending on what the robot was actually doing at that time.

Design dialogs to yield closure. This rule is related to the previous rule, *Offer informative feedback*, but specifically highlights the importance of providing a sense of finality after a series of actions has been carried out to achieve some goal. In other words, let users know they have accomplished what they set out to do.

We have applied an adaptation of this concept to the design of our progressions system in the RobotLink app. Progressions were specifically designed to be analogous to short-lived conversations by temporarily tracking the back-and-forth exchange of information as it relates to particular subject. This concept is intended to help users understand not only *what* a system is doing, but *why* it is behaving the way it is in situations where the system's behavior might otherwise be considered ambiguous. Additionally, the system can provide confirmation information to users, such as to acknowledge that they have successfully helped a robot after it requested their help.

Prevent errors. In its original context, this rule states that interfaces should be designed to prevent users from attempting to take actions that cannot be performed or otherwise putting the system into an error state. Shneiderman suggests techniques such as graying out menu items and only allowing users to input the correct kinds of information (such as numbers only for a phone number).

The RobotLink app is designed to be used by inexperienced or untrained users in situations that are far from ideal. It is entirely possible that the robot may have experienced a failure or may be performing a behavior that conflicts with the preferences of the person using the app. Thus, the RobotLink app was designed on the premise that something has already gone wrong with the robot and the

app's user is trying to understand, correct, or improve the situation. While error prevention should always be a priority in interface design, user errors under these circumstances should be assumed to make an already bad situation even worse. The app's design errs on the side of being conservative and makes no assumptions about which controls should be offered to the user. Instead, the robot is tasked with deciding which options are appropriate (i.e. correct) under the circumstances. Control options are presented to users as discrete options, allowing users to see all of their options at any given moment. There are no menu systems, adjustable values, direct controls (such as a joystick), or open ended inputs.

Strive for consistency. This rule originaly referred to using consistent terminology, color schemes, fonts, etc. across different aspects of the user interface. However, Shneiderman points out that this is the most frequently violated of his rules, due in part to the fact that there are many different forms of consistency. For example, the same sequence of actions should produce similar results each time they are performed.

We take a different perspective on consistency, focusing instead on the idea of providing bystanders with the opportunity to have a consistent experience across many different kinds of robots. The concepts of consistency and forming standards are the core principles underlying our development of both the robot state icons and the RobotLink app. In both cases, these interfaces were designed to be generic and flexible enough to be effectively used on a wide variety of robot platforms, with the eventual goal of becoming a standard universal interface for all robots — from quadcopter delivery drones to self-driving cars, and vacuum cleaners to assembly-line robots. While the exact information and prompts would vary, the consistent form-factor of the presentation and ubiquitous availability would help

users to remain grounded in a familiar experience, even when interacting with a type of robot they had never seen before.

Permit easy reversal of actions. Allowing users to easily "undo" their actions can reduce the stress and anxiety of using an interface, as this gives users the freedom to change their mind at a later point in time without having to worry about their previous actions having a permanent effect.

Bystanders may not have a choice whether or not they are co-located with an autonomous system. Thus, the RobotLink app attempts to give them some control over their situation. In keeping with this concept, commands and replies sent to the robot using the app have been designed to be easily reversed, and previously sent responses may be retroactively changed by the user within progressions at any time.

Support internal locus of control. "Internal locus of control" is a term taken from psychology, which generally refers to a person's belief that their actions are responsible for resulting events. Shneiderman originally used this term to describe how experienced users tend to desire a sense of control over the interface they are using. Here, we repurpose it to describe people's desire to have some input or control over situations in which they interact with a fully autonomous robot. This is supported to a limited extent (by design) though the RobotLink providing commands and responses that users can send to nearby robots to influence their behavior.

Reduce short-term memory load. This rule is based on observations of the limitations of human cognition, in particular that people can remember approximately 7 "chunks" of information in short term memory. Based on this constraint, Shneiderman recommends consolidating information displays and simplifying sequences of actions.

People using the RobotLink app do not need to hold anything in short-term memory in order to interact with any number of robots. Because of the ease with which the user can refer back to a machine's status page and check on a previous "conversation" with the robot, there is no load on short-term memory. On the other hand, if users want to look at the status of any of the other robots, they do not need to remember anything about them to do so. Users can easily refer back to the robot chooser list to match the robot's physical appearance and unique identifier to the machine in front of them.

6.1.2 Norman's Seven Principles

Donald Norman outlines seven principles in his book, *The Design of Everyday Things* [Norman, 2013], which are intended to help designers develop interfaces that are easily learnable and usable at first glance.

Discoverability. According to Norman, discoverability refers to allowing users to determine what kinds of actions they can perform given the current state of the device or interface through the use of constraints and affordances. The RobotLink app makes it possible to first identify which robot you are working with, and then quickly understand the available options for working with that particular machine once it has been selected. Only nearby robots are able to be selected.

Feedback. In Norman's seven stages of the action cycle, feedback is the information that helps people understand what happened as a result of their actions, such as what new state a system might be in. Both our state icons and the RobotLink app were designed specifically to expose the system's status to users. During the experiment, changes made to the system's state through the physical button in-

terface, RobotLink app, or automated decisions such as deciding when to return to its charger were all immediately reflected inside the app, including responses in the app to user actions (as a continuation of the "conversation").

Conceptual model. Norman asserts that when users are able to construct a good conceptual model of how the system works, they will also have a better understanding of it and have an increased feeling of control. It is possible that a bystander interacting with a fully autonomous robot may not understand the purpose of the machine or how it normally behaves, which can place serious limitations on their ability to acquire adequate level 2 and level 3 situation awareness [Endsley, 2004]. The robot state icons partially aid with this problem by helping people understand if the situation they are observing is "normal", while the RobotLink app can help people understand what a machine is doing and even how that behavior fits into a larger pattern of behaviors (e.g. a self-driving taxi waiting for a passenger, to transport them to a destination).

Affordances. Norman defines affordances as the relationship between properties of an object and the capabilities of the person or agent that determine how the object could be used. Norman specifies that *"affordances exist to make the desired actions possible."*

Rather than relying on physical affordances, the design of the RobotLink app allows robots to explicitly describe the services they provide and what they are currently doing. The app also provides discrete options to users, offering a finite set of possible requests for users to choose from.

Signifiers. This term is used by Norman to describe an indicator that signals what behaviors a person should take, and usually refers to discoverability and

user feedback in interface design. Signifiers communicate what behaviors are appropriate for people to take, and it is within this context that we focus on the communication of robot state to users. In particular, our robot state icons were designed to offer explicit signals to untrained users to help improve their situation awareness, and ultimately their decision making abilities.

Mappings. Mappings are defined by Norman as *"the relationship between controls and their actions . . . enhanced as much as possible through spatial layout and temporal contiguity."* In other words, mappings link the controls of an item to the item being controlled. The RobotLink app has been designed to provide information about the high-level behaviors running at any given time on the Robot. Furthermore, during our experiment, the command options presented to users through the app matched those available using the robots physical interface. Together, these design features created a direct match between the app and real world.

Constraints. Constraints guide users' actions by limiting the number of possible ways in which an object can be used. Rather than providing a complete set of master controls for every robot in the app, we chose to expose only controls relevant to the potential problems at hand for nearby machines. The app restricts the number of robots a user must choose from when connecting to a particular machine through the use of a low-power direct wireless connection. By providing the user with a minimal, yet complete, set of information needed to resolve the issue (rather than, for example, a large set of sensor output or other extraneous data), the app prevented unnecessary confusion and helped to promote a timely resolution of problems.

6.1.3 Nielsen's Heuristic Principles

Nielsen provided ten "heuristic" principles for interaction design [Nielsen, 1995], which are broad rules of thumb as opposed to specific guidelines. We provide Nielsen's description for each heuristic (given in italics), followed by examples of how we made use of them in our interface designs.

Visibility of system status. *"The system should always keep users informed about what is going on, through appropriate feedback within reasonable time."* This principle is highlighted by our state icons: to increase the visibility of a robot's status (which is usually otherwise opaque) and improve people's situation awareness.

Match between system and the real world. *"The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms."* Our robot state icons were crowdsourced to help make their meaning familiar to people without the need for explanation. The RobotLink app uses mappings between the real world and its interface in order to aid users during the "robot chooser" activity. Robots are represented by both an iconified representation of their hardware and by a unique identifier printed on the body of the robot and in the app. This helps users identify which machine in the app's list corresponds with each machine in the real world.

User control and freedom. *"Users often choose system functions by mistake and will need a clearly marked 'emergency exit' to leave the unwanted state without having to go through an extended dialogue. Support undo and redo."* We placed special emphasis in the RobotLink app on being able to make some limited form of control available for people such as bystanders who are not end users of the system but are rather forced to live alongside it.

Consistency and standards. *"Users should not have to wonder whether different words, situations, or actions mean the same thing."* As previously mentioned with Shneiderman's *Strive for Consistency*, the interaction methods designed in this dissertation emphasize being generic and applicable across a wide range of autonomous systems.

Error prevention. *"Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action."* As previously mentioned with Shneiderman's *Prevent Errors*, we place special emphasis on this area to "prevent making things worse than they already are".

Recognition rather than recall. *"Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate."* Robot state information is present on every screen and dialog box within the RobotLink app by making use of our robot state icons. Furthermore, anytime a user is working with a robot in the RobotLink app, the iconified depiction of the hardware platform and the uniquely identifying name printed on the body of the robot are displayed to disambiguate which machine's information is being viewed.

Flexibility and efficiency of use. *"Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions."* The RobotLink leverages a commonly used practice in app design of "remembering"

which display you were last viewing. This enables users to make fast context switches between various full screen applications on a smartphone without needing to reselect the robot they are working with each time the app is opened.

Aesthetic and minimalist design. *"Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility."* The RobotLink app uses a minimalist design based on Google's Material Design specification [Google, 2017]. Information was logically divided into "cards", and buttons used shadow to represent raised and lowered states which afforded pressing.

Help and documentation. *"Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large."* This heuristic is precisely the use case the RobotLink app was designed to support. The app assumes users have never seen or used the app before and strives to clearly present all nearby robots, their system states, and any options available to the user without the need for additional explanation.

Help users recognize, diagnose, and recover from errors. *"Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution."* This single heuristic principle best sums up the concept of creating failure-ready robots: help people (both users and bystanders) understand whether or not a system is working properly, guide them in making decisions about how they should handle these situations, and identify methods of

minimizing the effect the event will have on the people involved.

6.2 Failure-Ready Principles

Based on our literature research and experience, we are proposing a set of four guidelines and recommendations for creating failure-ready autonomous robots.

6.2.1 Provide Fast, Accurate, Situation Awareness

One of the problems with autonomous robots is that it can be very difficult for people to understand what they are doing, or whether they are even working correctly. This problem will only be made worse by placing these robots in public settings, where bystanders may unexpectedly find themselves needing to make a decision about how to interact with an intelligent machine they have never before encountered. Therefore it is critical that we identify methods of quickly and accurately providing nearby people with the situation awareness necessary to make informed decisions. This principal is supported by Nielsen's *Visibility of System Status* and *Recognition Rather than Recall* heuristic principles, as well as Shneiderman's golden rule of *Offer Informative Feedback*.

In this dissertation, this principle is supported both through the use of robot state icons and the RobotLink app. The robot state icons provide nearby people with a quick method of calibrating how they should perceive the robot; they immediately know if it is operating properly and whether or not it is safe to approach. However, the icons are not suited for communicating precise or detailed information. The RobotLink app, on the other hand, represents a potential method of communicating more complicated messages and is even capable of facilitating limited conversations, but the app comes with limitations of its own.

6.2.2 Support Users' Goals

Our work from Chapter 3 suggests that people's satisfaction with robots is tied to accomplishing their goals. That said, sometimes the *process* of achieving a goal is just as important or possibly even more important than arriving at an end state. The robot may be able to help mitigate situations in which it experiences a failure or becomes disabled simply by delegating the task it was performing to another (possibly human) agent or providing information which the end user may find helpful in improving their situation. For example, in the high-risk taxi scenario, the self-driving car calls for a human-driven taxi to come rescue the passenger with a ride to their destination after the self-driving car has broken down. The fact that a robot might not be 100% functional or even capable of performing the service it was designed to carry out does not disable it from continuing to be helpful.

6.2.3 Accommodate Bystanders

Explicit support for bystanders should be incorporated into the designs of a robot intended to be deployed in public settings where bystanders will be affected its behaviors. While the user's goals are naturally aligned with the goals of the robot, the needs and goals of bystanders may be very different. Another point to consider is that while a robot may operating "properly" with respect to its intended use case of performing a service for an end user, these same behaviors may not be well understood, appreciated, or even liked by some people.

One way in which bystanders can be supported is in the creation of ubiquitous secondary interface systems which could use to interact with a wide array of robots without needing to learn new interaction systems for every new robot encountered. This is concept is supported both by Shneiderman's *Cater to Universal Usability*

golden rule and Nielsen's *Consistency and Standards* heuristic principle.

Designers should also consider the circumstances in which bystanders might be interacting with the system, especially in the cases of publicly-deployed robots. Most of the people with which a publicly-deployed robot will interact with will not directly benefit from that particular robot's presence or operation; some of those people will have never used or ever benefited from the presence of similar robots in the past, and it is entirely possible that one or more of these people will be forced into a situation in which they must interact with or cooperate with the robot. In these situations, bystanders may appreciate being given some level of choice or control over the situation (Shneiderman's golden rule of Supporting Internal Locus of Control).

6.2.4 Ask for Help, but Don't Expect it

Research has shown that it is possible for robots to ask nearby people for help, though it doesn't always work. Hüttenrauch and Severinson Eklundh [2006] posited that for people to be willing and able to help a robot, they need to understand that the robot needs help, be in a position to help, and know how to provide the help needed. While people are willing to help robots, planning to recruit help from nearby people has not been found to be a highly reliable method. Offering people incentives such as candy has not shown to improve things either [Rosenthal et al., 2012]. Asking frequently for help should also be avoided, especially if it part of the robot's routine process of dealing with a commonly encountered problem. Rosenthal et al. [2012] found that people began to close their office doors in order to keep robots who asked for help on a regular basis away, and in another case Mutlu and Forlizzi [2008] found people were resentful of the amount they needed to help a robot.

Chapter 7

Conclusions and Future Work

With technologies such as self-driving cars and delivery drones poised for commercial introduction, autonomous robots will soon become a part of many people's daily life. Regardless of decades of research on failure prevention and reliability engineering, failures will inevitably continue to occur. When these robots fail, they will not only impact the people using them but potentially everyone else who happens to be around. The severity of these situations will be exacerbated by the fact that as a society we currently have no agreed upon method for the hundreds of millions of untrained people who will be expected to live and work alongside these unmonitored systems, and no way for these people to communicate with these robots during emergencies. For example, one of the results of poor situation awareness is that people sometimes perceive robots as failing when they are not, or as working properly when they failing. Lack of adequate or conflicting methods of communication between robots and bystanders will likely result in confusing, ambiguous, and frustrating situations. Accordingly, we should be developing strategies for dealing with these situations and identifying practical methods for communicating with publicly-deployed autonomous robots.

7.1 Contributions

This thesis took a human-centric approach to researching autonomous robot failures, investigating the effects robot failures have on people's perceptions of robotic services, developing a system for improving people's situation awareness around autonomous robots, and creating a smartphone-based interaction method designed to give people the ability to communicate with a wide array of different robots despite the lack of any training or prior experience.

REACTION scale: We developed a method for measuring people's responses to failures of autonomous robots that captured the main characteristics of failure situations while also highlighting the nuanced complexity present in these scenarios. The `REACTION` scale can differentiate between successful operations, different kinds of failures, and situations in which various recovery methods have been applied following a failure. We used this scale to perform an analysis of how failure severity, context risk, and different types of recovery strategies — specifically those based on supporting communication with people and those which directly support the completion of a task — influence people after a failure has occurred.

Icons communication paradigm: We developed an icon-based communication paradigm for communicating the state of autonomous robots with an aim to improve bystanders' situation awareness and then crowdsourced the design of a set of icons to test this concept. The icons were tested across 14 different robotic platforms with thousands of participants, and our results support the feasibility of using icons as a ubiquitous mechanism for improving people's situation awareness.

Smartphone-based interaction method: We designed a smartphone-based in-

teraction method, intended for use as a ubiquitous secondary interface for emergency and impromptu communication by untrained users. We implemented a functional prototype system, including a Bluetooth communication protocol and Android app called RobotLink, which were used to conduct an experiment. Our results indicated not only that participants were able to make use of the app to retrieve information about the robots and control them without any training, but also that the app made people feel more confident and in control. Use of the app also made the robots seem more predictable and increased participants' satisfaction with the robots.

Design principles for failure-ready robots: We identified four design principles which we believe will be instrumental to the creation of failure-ready robots. These principles are based on existing HCI design guidelines, findings from our research, and trends we have observed in literature.

7.2 Open Questions and Future Work

7.2.1 User Reactions to Failure

The study in Chapter 3 used a written story about a fictional character's (Chris) experience with an autonomous robot, in order to investigate how people felt about the system failing. The study then used their responses to construct a scale for measuring people's reactions to those events. A more mature and well tested version of the `REACTION` scale would be a valuable resource for understanding the impact of various failure scenarios, as well as evaluating the effectiveness of the recovery strategies designed for use in those situations.

The reaction scores computed in our analysis were based off of weighted values

that were derived by performing a separate exploratory factor analysis (EFA) on each collected data set. Although the EFAs produced similar factor weights in both story scenarios, we do not consider the scores to be directly comparable between the two. Furthermore, while the current questions were all written from the third-person perspective, the true value of this scale will be its ability to measure people's reactions to events that they experience personally. Accordingly, the next logical step will be the creation of a formalized survey-based measurement instrument designed to produce comparable results across different situations. The wording of the questions will require some modification to avoid context specific references (one of our study's shortcomings) and be written for a first-person perspective.

Finally, the scale will need to be tested using in-person experiments with physical robots, which cause participants to personally experiencing various failure scenarios and recovery strategies. The scale will need to be tested across multiple scenarios and with a variety of different hardware platforms to verify its use as a generic instrument.

Our investigation only targeted operator interactions with autonomous robots; however, robot failures will also impact bystanders. The ability to not only measure bystanders' reactions to failures, but also to robot behaviors in general, would be helpful in designing socially astute robot behaviors and detecting blundered social interactions.

7.2.2 State Notification Icons

Our work on developing a set of state icons as suitable method for autonomous robots to convey high-level messages to people was only intended to be the first step in a larger investigation. In this section, we discuss some of the subjects matters we plan to explore as we continue research in this area.

7.2.2.1 Experimentation using physical robots

The experiment described in Section 4.3 involved participants viewing images of robots that feature photoshopped icons. While this technique allowed us to quickly evaluate the feasibility of conveying messages through icons across thousands of participants, future work in this area would be better suited to having participants interact with physical robots featuring the icons. For example, how will people behave when confronted with a physical robot displaying a state icon? Will people shy away and keep their distance from robots displaying the `DANGEROUS` icon? Will they be more willing to place themselves in close proximity to a robot with a `SAFE` icon? Will changes in their behavior mirror the results of our analysis of robots' attributes (e.g. will the robots size affect the interpertation of the `DANGEROUS` icon)? How do peoples' prior experiences factor into their behavior, and to what extent can icons still be used to influence behavior? How will people react if they witness a change in the state icon? To what degree will additional contextual information such as lights, sounds, movement, or physical location effect peoples interpretations of the icons, both in conditions where the context clues should support the icon's message and where they would conflict with it?

7.2.2.2 Additional Icons

We have identified three additional messages that we are considering adding to our list of target messages. The target messages currently include icons that indicate if a robot is working properly (`OK`), needs help (`HELP`), is safe to be around (`SAFE`), is dangerous (`DANGEROUS`), or is turned off (`OFF`). The candidate messages being considered would express that the robot has experienced an error or failure (`FAIL`), that the robot is disabled (`DISABLED`), and that the robot has a notice or message it is trying to relay to a person (`MESSAGE`).

Non-dangerous error/failure (FAIL): None of our target messages would be appropriate in a situation where a robot has experienced a non-dangerous failure that prevents it from working. Furthermore, failed or broken robots may not necessarily be in need of help or be powered off. That would leave SAFE as the only remaining choice from our current set of target messages (though it is hardly appropriate). Thus, the current set of messages fail to adequately characterize the degraded condition of the robot in this situation and could benefit from this addition.

DISABLED: Another distinction needed is a variation on the message indicating that a robot has been physically powered off (OFF). A robot could be powered on and fully functional while still not being able to offer its services due to having been disabled (possibly for administrative reasons). There is also a clear difference between a robot having been disabled and having experienced a failure (although the two are not mutually exclusive). A robot may be powered on and experiencing a severe error, while still working well enough to limp itself out of the way of activity or even to return itself to a maintenance station (and therefore clearly not disabled). Both of these scenarios are distinct from a robot simply being powered off.

Notice/Message for a person (MESSAGE): An interesting scenario not adequately covered by the original set of target messages is how a robot would signal that it has a natural language message it needs to convey or is actively trying to communicate. The current design for the HELP icon implies that communication of some sort is involved, which is reasonable since the robot would need to provide sufficient information such that the person knows what to do to be helpful. However, there are many other scenarios in a which robot might need to signal

they are trying to deliver or convey information to a person, which do not involve requesting assistance.

7.2.2.3 Multiple Simultaneous Messages

There are many instances in which it would be useful to convey multiple target messages simultaneously. For example, a robot might require `HELP` and also want people to know that it is `SAFE` for them to approach. Alternatively, a machine could be inherently `DANGEROUS` for people to get too close to, while still working perfectly. Furthermore, it may be desirable to differentiate between a robot which has been intentionally disabled from one which has been disabled as a consequence of having experienced a failure, since that information might be helpful in predicting whether there is a chance the robot could be re-enabled in the near future.

It should be noted that some target messages are clearly mutually exclusive (e.g. `SAFE` and `DANGEROUS`), and that conveying these messages together would likely lead to confusion. That said, the case for being able to convey multiple target messages simultaneously is compelling, and it becomes even more so when we consider expanding the basic set of target messages. However, it is unclear what the most effective method of achieving this will be. Cursory research into this area suggests that it may be possible to display pairs of icons and have observers understand them as each conveying an individual message. Other possible strategies for conveying this information include the possibility of identifying icons that could inherently communicate multiple messages, or potentially even displaying individual icons in a rotation from a set of applicable icons.

7.2.3 Smartphone-based Interactions

We demonstrated that our smartphone-based interaction method is a plausible method to allow people to communicate with autonomous robots. However, our study was hardly comprehensive, with Section 5.5 listing several important limitations. In addition, there are several new aspects we intend to investigate in the future.

First, we would like to perform a study which more thoroughly investigates the use of the listed notification icons with our app, RobotLink. While we used some of the icons developed in Chapter 4 inside our app, we did not explicitly test for what people believed the icons meant, as they were being used as supplementary information to more verbose descriptions of the robot's status. We propose explicitly investigating how having state icons physically located on the robot compares with displaying them through the phone app. In particular, will displaying icons on the robot trigger *pull-style* interactions with the RobotLink app? When icons are displayed only in the app, do people interpret their meanings to be the same as when they are physically on the robots?

We also have several new ideas concerning app usage that have not been explored. Currently, the only method for selecting which robot to communicate with during *pull-style* interactions is to select the robot whose name and hardware drawing match the robot you are looking at. This is an effective method when there are only a few robots nearby. What then if there are a lot of robots around, all of which are of similar models and have similar names (e.g. a self-driving car lot)? We would like to implement and test four additional approaches to solving this problem.

The first looks at attempting to use signal strength to disambiguate which robot you are communicating with. Unfortunately, this approach was not practical to test

during our experiment since all of the robots were relatively close to each other (at most only a few feet apart). The second method uses Near Field Communication (NFC) tags to identify the robot. NFC is a form of close proximity RFID which has been used in "tap-to-pay" applications such as ApplePay and Google Wallet. This technology requires that the devices be separated by less than an inch and a half, and would therefore necessitate some sort of target for users to aim their phone at. An interesting disadvantage to this method is that it requires people to be relatively close to the target machine, which may be inconvenient or possibly even dangerous in some situations (such as if the robot is moving).

The third method uses QR codes on the robots which can be scanned using the phone's camera directly through the RobotLink app. While NFC tags can actually be used to trigger smartphone applications to automatically open, QR codes require a couple steps: opening the app and selecting the scanner, followed by aligning the camera to scan the code. Although the use of QR codes has an advantage, in that the users are not required to be as close to the target machine as NFC tags, people would still need to approach the machine to get close enough to scan the code.

The final method looks at using technology from Google's Project Tango to create an augmented reality viewport (similar to that in Pokemon Go). This would allow us to overlay information about nearby robots on top of the robots as the user looked at them through the app. This approach has the advantage of potentially being usable from a distance, with the added benefit of potentially even being able to "see" robots through walls. However, the Project Tango technology is still relatively new and has not been widely integrated into many devices yet.

One of the problems with testing smartphone interactions is that it requires us to provide users with a phone that has been preloaded with the RobotLink app. It

is not practically feasible for us to maintain enough versions of the app to use with a wide variety of phones, and many people still have older devices which do not have Bluetooth LE compatible chipsets. By supplying participants with a smartphone, we risk implying that they should be making use of it somehow, or possibly that they should not be (if we provide an excuse for giving them the phone such as "to track their movements"). Thus, it is difficult to test if participant's would ever think of trying to use their phone to communicate with the robots on their own, which is the stated purpose of *pull-style* interactions.

Therefore, we propose testing the basic premise of *pull-style* interactions by equipping robots with a combination QR code/NFC target that could be scanned, along with a short URL that could be easily typed into a web browser, and providing information about the robot's state through a web page. This would give participants three different methods for accessing the robot's information without initially biasing people into thinking about their phones. This testing method could then be expanded to investigate how various factor, such as the use of status icons or other context clues, influence whether or not people will use *pull-style* interactions to acquire more information about a robot.

7.3 Final Thoughts

This thesis examined autonomous robot failures through the lens of human-robot interaction. Research shows that failures by autonomous robots can have a serious impact on how people perceive these systems' utility. However, the work presented here also signals that the way in which those situations are handled can greatly influence the severity of such events. Finally, in view of the fact that autonomous robotic systems such as self-driving cars and delivery drones will soon become a

part of many people's daily lives, we investigated potential universal interaction methods which could be used by bystanders to gain situation awareness and successfully interact with many different kinds of fully autonomous systems despite lacking any training, prior knowledge, or experience with those platforms. In the near future there will be many people for whom an unanticipated bystander interaction with a fully autonomous robotic system in a public setting will be their first ever encounter with a robot of any kind, and it is our hope that this research will help contribute toward making those interactions positive experiences.

Bibliography

T. W. Andreassen. Antecedents to satisfaction with service recovery. *European Journal of Marketing*, 34(1/2):156–175, 2000. 3.1

B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009. 2.2.3

M. Baker, R. Casey, B. Keyes, and H. Yanco. Improved interfaces for human-robot interaction in urban search and rescue. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 3, pages 2960–2965. IEEE, 2004. 2.2.4

K. Baraka, S. Rosenthal, and M. Veloso. Enhancing human understanding of a mobile robot's state and actions using expressive lights. In *RO-MAN, 2016 IEEE*, 2016. 2.2.4

J. C. Becker and G. Flick. A practical approach to failure mode, effects and criticality analysis (fmeca) for computing systems. In *High-Assurance Systems Engineering Workshop, 1996. Proceedings., IEEE*, pages 228–236, Oct 1996. 2.3.2

A. J. Berinsky, G. A. Huber, and G. S. Lenz. Evaluating online labor markets for experimental research: Amazon.com's mechanical turk. *Political Analysis*, 20 (3):351–368, 2012. 3.4

C. L. Bethel and R. R. Murphy. Survey of non-facial/non-verbal affective expressions for appearance-constrained robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(1):83–92, 2008. 2.2.4

J. Bowie and D. Bowie. Perception of road signs by road users. In *Transportation Research Board 88th Annual Meeting*, number 09-1089, 2009. 2.2.4

C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin. Effects of nonverbal communication on efficiency and robustness in human-robot teamwork. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 708–713. IEEE, 2005. 2.2.4

D. J. Brooks, A. Shultz, M. Desai, P. Kovac, and H. A. Yanco. Towards state summarization for autonomous robots. In *2010 AAAI Fall Symposium Series*, 2010. 2.4.5

D. J. Brooks, C. Lignos, C. Finucane, M. S. Medvedev, I. Perera, V. Raman, H. Kress-Gazit, M. Marcus, and H. A. Yanco. Make it so: Continuous, flexible natural language interaction with an autonomous robot. In *Grounding Language for Physical Systems Workshop at the AAAI Conference on Artificial Intelligence*, 2012. 2.2.3, 2.2.4

D. J. Brooks, E. McCann, J. Allspaw, M. Medvedev, and H. A. Yanco. Sense, plan, triple jump. In *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6, May 2015. 2.2.3

R. Canham, A. H. Jackson, and A. Tyrrell. Robot error detection using an artificial immune system. In *Evolvable Hardware, 2003. Proceedings. NASA/DoD Conference on*, pages 199–207. IEEE, 2003. 2.3.2

J. Carlson and R. R. Murphy. How UGVs physically fail in the field. *IEEE Transactions on Robotics*, 21(3), 2005. 2.1.2, 2.4

E. Cha, A. Dragan, and S. Srinivasa. Perceived robot capability. In *24th IEEE International Symposium on Robot and Human Interactive Communication*, Kobe, Japan, Aug 2015. 1.1, 2.4.1, 3.1

E. Cha, M. Mataric, and T. Fong. Nonverbal signaling for non-humanoid robots during human-robot collaboration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 601–602, March 2016. 2.2.3, 2.2.4, 2.4.4

V. Chidambaram, Y.-H. Chiang, and B. Mutlu. Designing persuasive robots: how robots might persuade people using vocal and nonverbal cues. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 293–300. ACM, 2012. 2.2.4

K. Dautenhahn, S. Woods, C. Kaouri, M. L. Walters, K. L. Koay, and I. Werry. What is a robot companion - friend, assistant or butler? In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1192–1197, Aug 2005. 2.2.3

V. Denes-Raj and S. Epstein. Conflict between intuitive and rational processing: when people behave against their better judgment. *Journal of personality and social psychology*, 66(5):819, 1994. 2.2.5

M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco. Impact of robot failures and feedback on real-time trust. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction*, Tokyo, Japan, 2013. 1.1, 2.2.4, 2.2.5, 2.4, 2.4.1

M. R. Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 32, pages 97–101. SAGE Publications, 1988. 2.2.2

M. R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995. 3, 4

M. R. Endsley. *Designing for Situation Awareness: An Approach to User-Centered Design*. CRC Press, 2 edition, 2004. 6.1.2

M. R. Endsley and D. B. Kaber. Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, 42(3): 462–492, 1999. 2.2.3

C. Ferrell. Failure recognition and fault tolerance of an autonomous robot. *Adaptive behavior*, 2(4):375–398, 1994. 2.3.2, 2.3.3.2

K. Fischer, B. Soto, C. Pantofaru, and L. Takayama. Initiating interactions in order to get help: Effects of social framing on people's responses to robots' requests for assistance. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 999–1005, Aug 2014. 2.2.4

V. S. Folkes. Consumer reactions to product failure: An attributional approach. *Journal of Consumer Research*, 10(4):398–409, 1984. 3.1

A. Frutiger. *Signs and symbols: their design and meaning*. Van Nostrand Reinhold Company, 1989. 2.2.4

E. Gat et al. On three-layer architectures. *Artificial intelligence and mobile robots*, 195:210, 1998. 2.3.3.1

Google. Material design. https://material.io/guidelines/, April 20 2017. 5.1.2.2, 6.1.3

V. Groom, J. Chen, T. Johnson, F. A. Kara, and C. Nass. Critic, compatriot, or chump?: Responses to robot blame attribution. In *Proceedings of the 5th ACM/IEEE International Conference on Human-robot Interaction*, 2010. 2.4, 2.4.5

S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988. 5.3.6.3

Y. Hiroi and A. Ito. Asahi: Ok for failure: A robot for supporting daily life, equipped with a robot avatar. In *Proceedings of the 8th ACM/IEEE International Conference on Human-robot Interaction*, HRI '13, pages 141–142, Piscataway, NJ, USA, 2013. IEEE Press. 2.2.4, 2.4.2

H. Hüttenrauch and K. Severinson Eklundh. To help or not to help a service robot: Bystander intervention as a resource in human–robot collaboration. *Interaction Studies*, 7(3):455–477, 2006. 2.2.4, 2.4.4, 6.2.4

F. Ingrand, R. Chatila, and R. Alami. An architecture for dependable autonomous robots. In *IARP-IEEE RAS Workshop on Dependable Robotics*, Seoul, South Korea, 2001. 2.3.3.1

D. B. Kaber and M. R. Endsley. Out-of-the-loop performance problems and the use of intermediate levels of automation for improved control system functioning and safety. *Process Safety Progress*, 16(3):126–131, 1997. 2.2.3, 3, 4

M. R. Kabuka, S. Harjadi, and A. Younis. A fault-tolerant architecture for an auto-

matic vision-guided vehicle. *Systems, Man and Cybernetics, IEEE Transactions on*, 20(2):380–394, 1990. 2.3.2

P. Kaniarasu and A. Steinfeld. Effects of blame on trust in human robot interaction. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, Aug 2014. 2.4, 2.4.5

T. Kim and P. Hinds. Who should I blame? Effects of autonomy and transparency on attributions in human-robot interaction. In *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication*, 2006. 2.4, 2.4.5

R. A. Knepper, S. Tellex, A. Li, N. Roy, and D. Rus. Recovering from failure by asking for help. *Autonomous Robots*, 39(3):347–362, 2015. 2.2.3, 2.2.4, 2.4, 2.4.4

M. K. Lee, S. Kiesler, J. Forlizzi, S. Srinivasa, and P. Rybski. Gracefully mitigating breakdowns in robotic services. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*, March 2010. 1.1, 2.4, 2.4.3, 3.1, 3.1.1

B. Lussier, R. Chatila, F. Ingrand, M.-O. Killijian, and D. Powell. On fault tolerance and robustness in autonomous systems. In *3rd IARP-IEEE/RAS-EURON Joint Workshop on Technical Challenges for Dependable Robots in Human Environments*, pages 351–358, Sept 2004. 1.1, 2.1.1, 2.3, 2.4

R. R. Lutz and R. Woodhouse. Bi-directional analysis for certification of safety-critical software. In *1st International Software Assurance Certification Conference (ISACC'99)*, 1999. 2.1.3

C. Matuszek, B. Mayton, R. Aimi, M. P. Deisenroth, L. Bo, R. Chu, M. Kung, L. LeGrand, J. R. Smith, and D. Fox. Gambit: An autonomous chess-playing

robotic system. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4291–4297. IEEE, 2011. 2.2.3

C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to parse natural language commands to a robot control system. In *Experimental Robotics*, pages 403–415. Springer, 2013. 2.2.3

J. P. Mendoza, M. Veloso, and R. Simmons. Plan execution monitoring through detection of unmet expectations about action outcomes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3247–3252. IEEE, 2015. 2.3.3.1

N. Moray, T. Inagaki, and M. Itoh. Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, 6(1):44–15, 2000. 2.2.5

M. W. Mueller and R. D'Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 45–52, May 2014. 2.3.3.2

R. R. Murphy and D. Hershberger. Handling sensing failures in autonomous mobile robots. *The International Journal of Robotics Research*, 18(4):382–400, 1999. 2.1.2, 2.3.2, 2.3.3.1

B. Mutlu and J. Forlizzi. Robots in organizations: The role of workflow, social, and environmental factors in human-robot interaction. In *Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction*, HRI '08, pages 287–294, New York, NY, USA, 2008. ACM. 2.4.4, 6.2.4

J. Nielsen. Usability inspection methods. In *Conference companion on Human factors in computing systems*, pages 413–414. ACM, 1994. 6.1

J. Nielsen. 10 usability heuristics for user interface design. https://www.nngroup.com/articles/ten-usability-heuristics/, Jan 1995. 6.1.3

R. E. Nisbett, C. Caputo, P. Legant, and J. Marecek. Behavior as seen by the actor and as seen by the observer. *Journal of Personality and Social Psychology*, 27(2):154, 1973. 3.4

D. A. Norman. *The design of everyday things: Revised and expanded edition*. Basic books, 2013. 2.2.1, 2.2.2, 6.1, 6.1.2

D. R. Olsen and M. A. Goodrich. Metrics for evaluating human-robot interactions. In *Proceedings of PERMIS*, volume 2003, page 4, 2003. 4

D. R. Olsen, Jr. and S. B. Wood. Fan-out: measuring human control of multiple robots. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2004. 2.2.3, 3, 4

D. W. Payton, D. Keirsey, D. M. Kimble, K. Jimmy, and J. K. Rosenblatt. Do whatever works: A robust approach to fault-tolerant autonomous control. *Applied Intelligence*, 2(3):225–250, May 1992. 1.1, 2.1.1, 2.3.3.1

C. Pecheur. Verification and validation of autonomy software at nasa. 2000. 2.3.1, 2.3.1.3

H. Pentti and H. Atte. Failure mode and effects analysis of software-based automation systems. *VTT Industrial Systems, STUK-YTO-TR*, 190:190, 2002. 2.3.1, 2.3.1.1

M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009. 2.1.3

I. Rae, L. Takayama, and B. Mutlu. The influence of height in robot-mediated communication. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 1–8. IEEE Press, 2013. 4.3.3.4

L. D. Riek, T.-C. Rabinowitch, P. Bremner, A. G. Pipe, M. Fraser, and P. Robinson. Cooperative gestures: Effective signaling for humanoid robots. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 61–68. IEEE, 2010. 2.2.4

M. L. Riemer-Reiss and R. R. Wacker. Factors associated with assistive technology discontinuance among individuals with disabilities. *Journal of Rehabilitation*, 66 (3):44, 2000. 2.2.5

A. Rosenfeld, N. Agmon, O. Maksimov, A. Azaria, and S. Kraus. Intelligent agent supporting human-multi-robot team collaboration. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 1902–1908. AAAI Press, 2015. 2.4.2

S. Rosenthal, M. Veloso, and A. K. Dey. Is someone in this office available to help me? *Journal of Intelligent & Robotic Systems*, 66(1):205–221, 2012. 2.4, 2.4.4, 6.2.4

J. Scholtz. Theory and evaluation of human robot interactions. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2003. 2.2.1, 4

T. B. Sheridan and W. L. Verplank. Human and computer control of undersea teleoperators. *MASSACHUSETTS INST OF TECH CAMBRIDGE MAN-MACHINE SYSTEMS LAB*, 1978. 2.2.3

B. Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Pearson Education India, 2010. 6.1, 6.1.1

P. Slovic and E. Peters. Risk perception and affect. *Current directions in psychological science*, 15(6):322–325, 2006. 2.2.5

G. Steinbauer. A survey about faults of robots used in robocup. In *RoboCup 2012: Robot Soccer World Cup XVI*, pages 344–355. Springer, 2013. 2.1.2, 2.4

K. S. Suh. Impact of communication medium on task performance and satisfaction: an examination of media-richness theory. *Information & Management*, 35(5): 295–312, Mar. 1999. 2.2.4.1

D. Szafir, B. Mutlu, and T. Fong. Communicating directionality in flying robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 19–26. ACM, 2015. 2.2.4, 2.4.2

L. Takayama, W. Ju, and C. Nass. Beyond dirty, dangerous and dull: what everyday people think robots should do. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, pages 25–32. ACM, 2008. 2.2.3

V. Verma. Anecdotes from rover field operations. *RIACS Summer student research program report-NASA Ames research center*, 2001. 2.1.2

P. S. Visser, J. A. Krosnick, and P. J. Lavrakas. Survey research. In H. T. Reis and C. M. Judd, editors, *Handbook of research methods in social and personality psychology*, volume xii, chapter 9, pages 223–252. Cambridge University Press, New York, NY, US, 2000. 3.4

H. A. Yanco and J. L. Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, pages 111–119, 2002. 2.2

H. A. Yanco and J. L. Drury. Classifying human-robot interaction: an updated taxonomy. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, Oct 2004. 2.2.1, 4

H. Yasuda and M. Matsumoto. Psychological impact on human when a robot makes mistakes. In *System Integration (SII), 2013 IEEE/SICE International Symposium on*, pages 335–339, Dec 2013. 2.4.4

Appendix A

Smartphone App Experiment

A.1 Experimenter Script

`[When participant arrives, after introducing yourself]`

Thank you for participating in our study. I'm going to be reading instructions to you from this script. If you have any questions feel free to ask and I will do my best to answer them. Please read through this informed consent form carefully and sign at the bottom. There is also a video consent form you need to sign giving us permission to record video and audio in the room. If you have any questions feel free to ask me at any time ... During this study, you will be asked to complete several questionnaires and perform a task in which you interact with our robots twice. The entire process should take about 45 minutes, and if you complete it you will be compensated up to $15 in the form of Amazon gift cards. You will receive at least $5 for simply completing the entire study, and have the opportunity to earn up to an additional $10 based on your performance.

`[Once informed consent and video release forms are filled out]`

Before I explain the activity, please fill out the survey on the machine in front of you. . . . Thanks.

`[After the survey has been completed]`

The task we are asking you to perform today is a kind of game that consists of two parts. The first thing you will need to do is play a video game on this Android smartphone. Pressing this button turns the screen on. Taping on this icon will take you to the game *(tap icon to demonstrate)*.

In the game, you will see balloons floating up the screen. You can score points from "popping" the balloons by tapping on them. However, every time you pop a purple balloon you will lose 30 points. The points you get in the game count towards your overall score. The more points you score, the more money you will receive for completing the study. That said, the video game can only be played while you are inside this special marked off area. If you leave this area, you will not be able to continue popping balloons until you return. There is a clock on the screen in front of you, which shows the same time as the time displayed in the upper right hand corner of the game. You have until time runs out to score as many points as possible. You need to keep the phone with you at all times during the experiment, even when you are not playing the game. Please do not set the phone down unless I let you know it is ok to do so. If you forget, I will try to remind you.

The second thing you need to do is use the robot vacuum cleaners over here to clean up these beads I will be scattering on ground. While you earn points for work in the video game, you can loose them for having beads left on the ground. The amount of points you get to keep from the balloon game will be based on the percentage of the beads collected by the robots. Only the robots can collect the

beads, you are not permitted to pick up any of the beads yourself or enter the framed area the robots are inside. You are, however, allowed to reach inside to touch the robots. The robots can be running while you are over here playing the balloon game. You can start the robots by pressing the button that says "clean" (or "start", depending on the brand).

You will have 6 and a half minutes during which you need to both play the balloon game and have the robots collecting beads. It is up to you to keep track of the time remaining. All the robots need to be back on their chargers before time runs out. Each robot that is not on its charger will cost you 100 points. They are programmed to automatically return to their chargers after a short period of time, but will not necessarily return before time expires. You may manually return robots to their chargers by reaching across the edge of the frame to retrieve a robot, but only if it is within arms reach.

Sometimes the robots' dustbins become full and need to be emptied. If you need to empty a robot's dust bin, make sure to carefully pour the collected beads into one of these red containers (show container) so they can be counted later. Let me quickly show you how to empty the dustbins. (demonstrate)

Some of our robots work better than others. If for some reason you have trouble getting a robot to work you can try to fix it, but we cannot guarantee that all the robots will work and I cannot help you fix any problems. While you must use the robots to collect the beads, you are not required to use all three of them. You may not disassemble the robot (except for emptying the dust bins), or take out any of the robots' batteries.

You will be doing this activity twice. Prior to the start of each run, you must be in this area behind the table. After each run we will ask you take a short questionnaire while we reset the game. We will also swap out two of the robots

with these two other robots between runs. Breaking any of these rules results in a loss of half your points. Your final score will be calculated using your balloon game score multiplied by the percentage of beads collected by the robots, minus 100 points for each robot that is not on its base station at the end of the game. Your compensation will be based on the higher of your two final scores. If you need to take a break at any time during the experiment, please let me know.

Do you have any questions?

`[Before the run with the smartphone support turned on]`

One of the things we are testing in this study is a new technology that allows robots and nearby smartphones to communicate with each other. In addition to the balloon game this phone has also been loaded with an app you can use to see and control nearby robots, as well as allow the robots to send you information. (show app on phone) During this run we will be enabling this technology. Once the runs starts, the robots will appear in the app. To switch between apps, either press the circle or the square at the bottom of the screen. To go back to a previous page inside an app, press the triangle in the bottom left hand corner.

`[Second Run]`

Ok, we are going to do the same thing again. I have reset the time clock and game score, and swapped out two of the robots. (go to either smartphone text above, or conditions 1and 2 below)

`[Conditions 1 and 2, Second run]`

However, in this run the app that lets the robots talk to the phone has been disabled.

`[After each run]`

Ok, time is up. Please use the computer in front of you to fill out the questionnaire while I reset the robots and compute your score.

`[At end of experiment]`

Thank you for participating in our study. Here is your Amazon giftcard. I need to mention that these robots have all been modified, causing their behaviors to be slightly different from those of the original products. Do you have any final questions?

A.2 Questionnaires

A.2.1 Pre-Experiment Questionnaires

How did you find out about our study? Please select all that apply.

– *Email advertising the study*

– *An informational flyer on campus*

– *Recommended by a friend*

– *Other, please specify.*

Have you previously participated in a robot related study at UML or another university?

Yes, No **If yes, please elaborate.**

Has anyone who previously participated in this study discussed their experience with you?

Yes, No **If yes, please elaborate.**

Have you previously witnessed robots operating inside this building since the beginning of the year?

Yes, No **If yes, please elaborate.**

Have you received any advice or recommendations related to participating in this study?

Yes, No **If yes, please elaborate.**

Please indicate your level of agreement with each of the individual statements regarding risk-taking activities.

I like to test myself ever now and then by doing something a little risky.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Sometimes I will take a risk just for the fun of it.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I sometimes find it exciting to do things for which I might get into trouble.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Excitement and adventure are more important to me than security.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Please indicate your level of agreement with each of the following statements about technology.

Technology makes life easy and convenient.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology makes life complicated.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology gives people control over their daily lives.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology makes people dependent.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology makes life comfortable.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology makes life stressful.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology brings people together.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology isolates people.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology increases personal safety and security.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Please indicate your level of agreement with each of the following statements describing yourself

I like to keep up with the latest technology.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I generally wait to adopt a new technology until all the bugs have been worked out.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I enjoy the challenge of figuring out high tech gadgets.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I feel confident that I have the ability to learn to use technology.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Technology makes me nervous.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

If a human can accomplish a task as well as technology, I prefer to

interact with a person.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I like the idea of using technology to reduce my dependence on other people.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Please provide us with your level of experience in the following areas. Rate the following statements as they apply to you.

I have seen real robots in person.

Never, Rarely, Regularly, Frequently

I have seen real robots on TV.

Never, Rarely, Regularly, Frequently

I have operated or programmed a robot before.

Never, Rarely, Regularly, Frequently

I have used a smartphone.

Never, Rarely, Regularly, Frequently

I have played games on a smartphone.

Never, Rarely, Regularly, Frequently

I have written my own smartphone app.

Never, Rarely, Regularly, Frequently

I have personally jailbroken, rooted, or installed a custom ROM onto my own smartphone.

Never, Rarely, Regularly, Frequently

I have written computer software.

Never, Rarely, Regularly, Frequently

I have written robotics software.

Never, Rarely, Regularly, Frequently

I have used a robot vacuum cleaner.

Never, Rarely, Regularly, Frequently

I currently own or have previously owned the following devices:

– *An Android Smartphone*

– *An Android Tablet*

– *An iPhone*

– *An iPad*

– *A Windows Smartphone*

– *A Windows Surface Tablet*

– *A Blackberry*

– *A PDA device, such as PalmPilot*

– *Other "smart" device, please specify:*

– *iRobot Vacuum Cleaner*

– *Neato Vacuum Cleaner*

– *Other robotic vacuum cleaner, please specify:*

Please answer the following questions about your attitude towards robots (in general).

I would feel uneasy if I was given a job where I had to use robots.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

The word "robot" means nothing to me.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I would feel nervous operating a robot in front of other people.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I would hate the idea that robots or artificial intelligences were

making judgements about things.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I would feel very nervous just standing in front of a robot.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I would feel paranoid talking with a robot.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

A.2.2 Post-Run Questionnaire

The first page of the post run questionnaire will measure the participants situation awareness during the run.

How many robots were you able to make use of in cleaning the floor?

0, 1, 2, 3

How many of the robots that were able to clean the floor required or requested your assistance in order for them to start or continue cleaning (not counting your initial instruction to begin cleaning)?

0, 1, 2, 3

What kind of help did the robot(s) require or request? Select all that apply.

– *Be reset (power cycled)*

– *Have dust bin emptied*

– *Be unplugged.*

– *Dust off sensors*

– *Free jammed sweeper brush*

My satisfaction level with the robots was _____.

Very Low, Low, Moderately Low, Acceptable, Moderately High, High, Very High

My confidence level with regard to understanding what each robot was doing was _____.

Very Low, Low, Moderately Low, Acceptable, Moderately High, High, Very High

It was _____ to determine what I needed to do to make each robot work.

Very Difficult, Difficult, Moderately Difficult, Neither easy not difficult, Moderately Easy, Easy, Very Easy

The robots actions were surprising or unpredictable.

Very Disagree, disagree, somewhat disagree, neutral, somewhat agree, agree, very agree

This page measures participant workload.

How mentally demanding was the task?

Very Low, Low, Somewhat Low, Neutral,Somewhat High, High, Very High

How physically demanding was the task?

Very Low, Low, Somewhat Low, Neutral,Somewhat High, High, Very High

How hurried or rushed was the task?

Very Low, Low, Somewhat Low, Neutral,Somewhat High, High, Very High

How successful were you in accomplishing what you were asked to do?

Very Unsuccessful, Unsuccessful, Somewhat Unsuccessful, Neutral,Somewhat Successful, Successful, Very Successful

How hard did you have to work to achieve your level of performance?

Very Low, Low, Somewhat Low, Neutral,Somewhat High, High, Very High

How insecure, discouraged, irritated, stressed, and annoyed were you?

Very Low, Low, Somewhat Low, Neutral,Somewhat High, High, Very High

The following pages will only be given to participants after the run where smartphone communication is supported, and asks questions about where the participant believes the information they received on the smartphone originated from.

I made use of the phone app to control the robots and get information about them.

Never, rarely, sometimes, about half the time, frequently, most of the time, always

The information about the robot(s) I saw on the phone came directly from the robot(s).

Very Disagree, disagree, somewhat disagree, neutral, somewhat agree, agree, very agree

The information about the robot(s) I saw on the phone came from a remote system controlling the robot(s).

Very Disagree, disagree, somewhat disagree, neutral, somewhat agree, agree, very agree

The information about the robot(s) I saw on the phone came from another person.

Very Disagree, disagree, somewhat disagree, neutral, somewhat agree, agree, very agree

I could tell which robot the information displayed on the phone was coming from.

Never, rarely, sometimes, about half the time, frequently, most of the time, always

It was _____ to control the robots using the phone app than to use the buttons on the robots.

Much Easier, Easier, Somewhat Easier, About the Same, Somewhat Harder, Harder, Much Harder

It was _____ to tell what I needed to do with the robots by looking at the phone app than by looking at the actual robot.

Much Easier, Easier, Somewhat Easier, About the Same, Somewhat Harder, Harder, Much Harder

It was _____ to figure out how to use the phone to get information about and control the robots.

Very Easy, Easy, Somewhat Easy, Neither easy or hard, Somewhat Hard, Hard, Very Hard

A.2.3 Post-Experiment Questionnaires

I felt more in control of the robots during the run in which the robots could communicate over the smartphone.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

The robots sending messages to the phone were disruptive and distracting.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Receiving information from the robots on the phone was helpful and informative.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

I like having the ability to communicate with the robots over the phone.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

The messages sent by the robots to the phone were confusing.

Strongly Disagree, Disagree, Somewhat Disagree, Neutral, Somewhat Agree, Agree, Strongly Agree

Do you have any recommendations for how the phone app could be improved?

The last page of the questionnaire collects demographic information.

Please answer the following demographics questions.

- Age

- Gender
  - *Male*
  - *Female*
  - *Other*
  - *Prefer not to answer*
- What is the highest degree or level of education you have completed?
  - *Some high school, no diploma*
  - *High school or equivalent (GED)*
  - *Some college, no degree*
  - *Trade/technical/vocational training*
  - *Associate Degree*
  - *Bachelor's Degree*
  - *Master's Degree*
  - *Doctorate Degree*