Developing an HRI Graduate Course Project on Usability Testing Using the Baxter Robot

Sarah Zhou*, and M. L. Cummings†

Humans and Autonomy Laboratory, Duke University, Durham, NC, 27708, USA

The growing interest in autonomous humanoid robots in industrial settings have pushed for the development of theoretical and methodological usability testing frameworks in academia for human-robot interaction (HRI) that will improve and standardize the way humans are interacting with these robots as well as their construction. This study presents the details of the design, implementation, and results of a project developed to teach HRI usability testing for a graduate level Mechanical Engineering course on the fundamentals of HRI. The report presents the authors evaluation of the effectiveness of the project and the Baxter robot in teaching HRI usability testing based on student work and feedback, and ways to improve this project for future educational purposes. As a conclusion, the limitations in robot-arm precision and the error-prone environment setup may have decreased student enthusiasm for working with the Baxter robot or cobots in general.

I. Introduction

Human-Robot Interaction (HRI) is a growing branch of robotics research and application involving the understanding, design and evaluation of robotics systems that will interact with humans, particularly semi or fully autonomous robots that can work alongside humans [1–3]. One fundamental ”hard problem” of HRI today is human supervisory control of robots for routine industrial tasks (ie. picking and placing, welding, etc.) such that human operators are required for the functions of supervisory control of these ”telerobots” [4]. Although the growing interest in semi-autonomous humanoid telerobots in industrial settings have strengthened the development of human-robot collaboration, it is still a relatively new field, and limited such standardized usability testing practices and theories exist today.

As demand for robots in industry has risen, demand for HRI university courses working with robots has increased as well [5, 6]. And as a broad field combining technical with human factors with no standardized curriculum, the dilemma facing professors and educators today is choosing the most effective education tools, topics, methods, and projects to teach them HRI fundamentals, especially considering the broad spectrum of backgrounds that students interested in HRI may come from.

Important factors that should be addressed in these frameworks include safety, usability, social acceptance, user experience, and societal impact with the ultimate goal of assessing the level of trust and acceptance for the robot in society and industry [7]. In this way, incorporation usability testing in the HRI curriculum builds an important bridge between HRI education, academia, and industrial applications.

This paper presents the details of the design, implementation, and results of a project developed to teach robot usability testing for a graduate level Mechanical Engineering course on the fundamentals of human-robot interaction. The students are mostly graduate students; However, a couple undergraduate seniors are also registered under special permission that is granted if they meet the basic engineering fundamentals requirement. The project will involve conducting a usability evaluation of the commonly-used collaborative robot (cobot) Baxter Research Robot in academia, developed by Rethink Robotics in 2012, to do a pick and place task. The goal is to present and introduction to the theory, methods, and tools of usability testing for the students to conduct their own. In this report, we evaluated the effectiveness of Baxter in teaching usability testing and achieving our other learning objectives (see Methods section), and its potential in the field of HRI education. We also investigated ways to improve this project for future educational purposes based on student outcomes.

The rest of this paper is organized as follows. Section II provides the background of usability evaluation in HRI studies. Section III describes the methods of conducting such an HRI study, while Section IV presents the results collected from students. Section V concludes this paper with a detailed discussion and potential future directions.

*Undergraduate Student, Computer Science (BS ’19)

†Professor, Department of Electrical and Computer Engineering, 304 Research Dr, Durham, NC 27708

II. Background

Usability testing, which has been widely used in various fields, evaluates systems by testing on users or operators [8–10]. Especially in human-robot interaction related applications, usability testing can provide implications about the effectiveness of the collaboration between human operators and autonomous systems [11, 12]. We provided reference papers to the students introducing robot usability testing frameworks and metric selection. This section summarizes those studies and how they are relevant to the learning objectives of the assignment. This section also reviews the Baxter robot and its advantages and disadvantages compared to alternatives.

A. Reference Papers

USUS Evaluation Framework

The USUS evaluation framework is a theoretical and methodological evaluation framework developed from a human-centered HRI perspective for evaluating usability, social acceptance, user experience, and societal impact of humanoid robots used in collaborative tasks. Its primary motivation is to evaluate the extent to which users accept the robot as both a support for cooperative work and as part of society–presenting an humanistic view on evaluating humanoid robots [7].

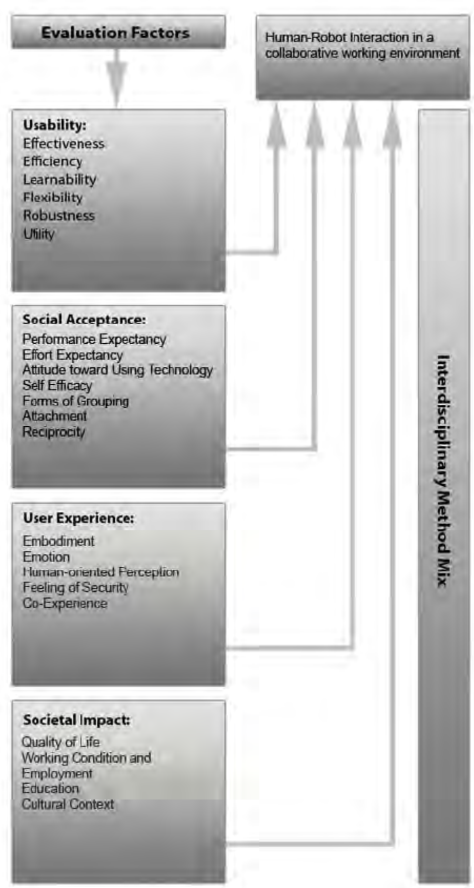

Figure 1. The USUS evaluation framework for user-centered HRI [7]

The framework consists of two parts: 1) a theoretical framework defining the relevant evaluation factors and indicators combined with (2) a methodological framework explaining the methodological mix of quantitative and qualitative measures to address each these factors. As figure 1 depicts, the theoretical factor model is a multi-level indicator model targeting usability, social acceptance, user experience, and societal impact [7]. All factors chosen for evaluation are measure through a methodological mix to be assessed during an iterative design process of human-robot collaboration. Some examples of measures in the methodological mix are expert evaluation, user studies, questionnaires, physiological measures, focus groups, and interviews.

System Usability Scale (SUS)

The SUS scale is a simple ten-item scale giving a global view of subjective assessments of usability [13]. John Brooke, the developer of the scale, defines usability in his paper as a system's general "appropriateness to a purpose of any particular artefact". It was originally developed by Brooke in 1986 to be a "quick and dirty", low-cost but reliable scale to measure the usability of electronic office systems [13]. However, now (more than thirty years later) this scale is applied to a wide variety of products and services, including hardware, software, mobile devices, website and applications. It has even been defined in the International Organization for Standardization (ISO) under ISO 9241-11 to further clarify correct ergonomics of human-system interaction.

The scale consists of 10 statements that are then rated by the user on a 5-point Likert scale ranging from "Strongly Disagree" to "Strongly Agree" (see Figure 2 below).

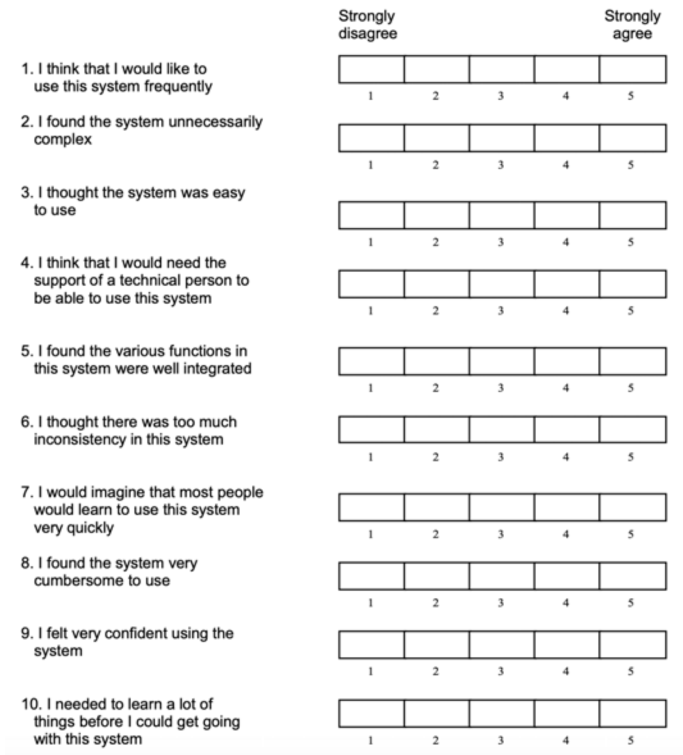

Figure 2. The SUS scale is a simple ten-item scale giving a global view of subjective assessments of usability. [13]

Selecting Metrics to Evaluate Human Supervisory Control Application

Another reference paper assigned was on the development of a methodology selecting metrics to evaluate human supervisory control applications [14]. Human supervisory control emphasizes that the human operator, acting from a comfortable nearby space environment, supervises a lower level intelligence teleoperator by intermittently monitoring and reprogramming as necessary, as opposed to fully autonomous robot. The methodology is based on 1) cost-benefit analyses in which each metric set has advantages, limitations, and costs, thus the added value of different sets for a given context can be calculated to select the set that maximizes value and minimizes costs, and 2) generic metric classes in which metrics are mission-specific, but metric classes are generalizable across different missions.

Five main collaborative metrics are identified that apply to supervisory control teams composed of humans and autonomous platforms: 1) mission effectiveness, 2) autonomous platform efficiency, 3) human behavior efficiency, 4) human initial state (behavior precursors), and 5) collaborative metrics. These metrics aim to address the collaboration between the human and the autonomous platform he is controlling, collaboration among humans that compose the team, and autonomous collaboration among platforms.

The paper concludes that which specific metrics should be used to evaluate a system will depend on many factors, but as a rule-of-thumb, it is proposed that at a minimum, one metric from each class should be used to provide a multi-dimensional assessment of the human-automation team. In addition, the paper provides a guideline in the form of a preliminary list of evaluation criteria for selecting supervisory-control metrics, as well as a criteria list that includes experimental constraints, construct validity, comprehensive understanding gained, statistical validity and efficiency, and appropriateness of the measuring technique [14].

An Active-Learning Approach to Mobile Robotics Education

Laurel D. Riek presented an upper-level undergraduate/graduate course in the Computer Science and Engineering Department of the University of Notre Dame titled Autonomous Mobile Robots [15]. The course emphasized the following instructional approaches that are positively regarded in the engineering education literature and well suited to mobile robotics instruction: 1) active learning, with students immediately practicing concepts learned in lectures on the robots; this helps them engage with the course material, 2) the course featured cooperative learning, with students working on assignments together in small teams, and 3) problem-based learning, where all lectures and assignments brought students back to the fundamental computational problems in mobile robotics.

The robotic platform students used in the course was the Turtlebot, an open hardware/open software platform comprised of an iRobot Create wheeled platform, a Microsoft Kinect sensor, and an laptop running Ubuntu Linux, and the Robot Operating System (ROS).

While a number of students encountered unexpected behaviors in the software and hardware, students performed well in the course overall and rated it an enjoyable experience. In the qualitative course evaluation, students were given a pretest on the first day of class before the first lecture began or any course materials had been distributed, and a post-test on the last day of class. The tests were identical and had the following questions: Q1) What is a robot? Q2) What are robots capable of? and Q3) What does it mean for a robot to be autonomous?. For all three questions, all students provided more detailed answers in their post-tests compared to their pretests, demonstrating effective learning in the course. Their post-test answers included specific examples and definitions from topics discussed in class and a clearer understanding of the capabilities and limitations of robots, and the different categories of autonomy (e.g., mixed initiative control) [15].

B. Baxter Robot and Rethink Robotics

Baxter is a humanoid, anthropomorphic robot with two seven degree-of-freedom arms and sensing technologies, including force, position, and torque sensing and control at every joint, cameras in support of computer vision applications, a training cuff that allows zero-gravity positioning of the limbs, gripper, and integrated user input and output elements such as a head-mounted display, and buttons (as shown in Figure 3 below) [16].

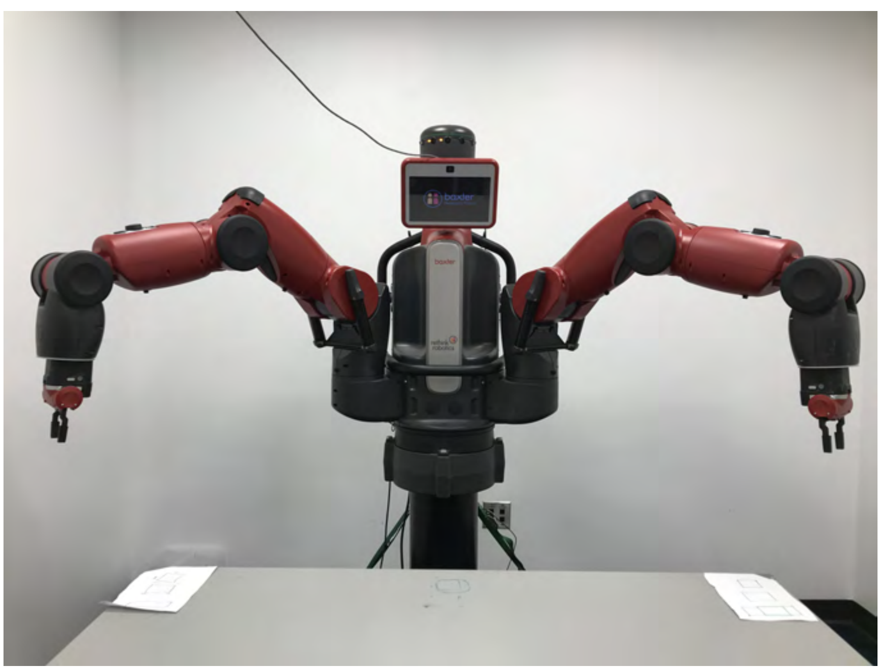

Figure 3. Baxter Research Robot in neutral position

There are two versions of Baxter, Baxter for Manufacturing and the Baxter Research Robot. Baxter Manufacturing Robot is pre-installed with Intera software, allowing Baxter to be taught tasks programming-free through its task creation manager feature [16]. However, for this project, we used the Baxter Research Robot, which comes installed with the SDK, allowing researchers to write custom software for Baxter. The main way to interface with the research bot will be programmatically using Baxter Interface libraries available through ROS, an open source framework for writing robot software across a wide variety of robotic platforms [17].

The Baxter robot was chosen as the project vehicle because Rethink Robotics, the makers of Baxter, recently filed for bankruptcy on October 3, 2018, and this is an important piece of information for the usability study.

One major drawback of the Baxter Robot in manufacturing was the lack of precision in its movements due to its force sensing capability implemented using series elastic actuators (SEAs) in their joints. The SEAs introduce substantial flexibility in the joints of the robot, which is good for safety, but bad for precision and motion performance due to how difficult it is to control a flexible manipulator [18]. However, while Baxter was not well-suited for industrial users, the consensus is it was an appropriate solution for universities and academia due to its safety, simple multimodal user interface, market penetration in academia, and ROS support.

III. Methods

The assignment has six parts: 1) Ubuntu installation in virtual machine, 2) ROS installation and workstation setup, 3) connecting to Baxter, 4) pick and place with main mode and demo mode, 5) usability test based on reference papers, and 6) survey to evaluate the effectiveness of the assignment and Baxter as an educational tool. The full assignment instruction is shown in Appendix A.

A. Learning Objective

Students will learn current theoretical and methodological usability testing HRI frameworks by reading their assigned case studies, and learn to conduct their own on a simple pick and place task using the Baxter research robot. They will also familiarize themselves with current news and trends surrounding cobots in industry and education, and learn to work with ROS, the most widely used robot development framework today. Lastly, since it is an introductory class, we hope this assignment will increase students interest in HRI and working with cobots, as well as bolster their enthusiasm for the integration and applications of cobots in society.

B. Requirements of Students

Students will be turning in: 1) three videos of Baxter performing pick and place task through main mode, 2) the Python file written for the pick and place task, 3) three videos of Baxter performing pick and place task through demo mode, and 4) final report in research paper format.

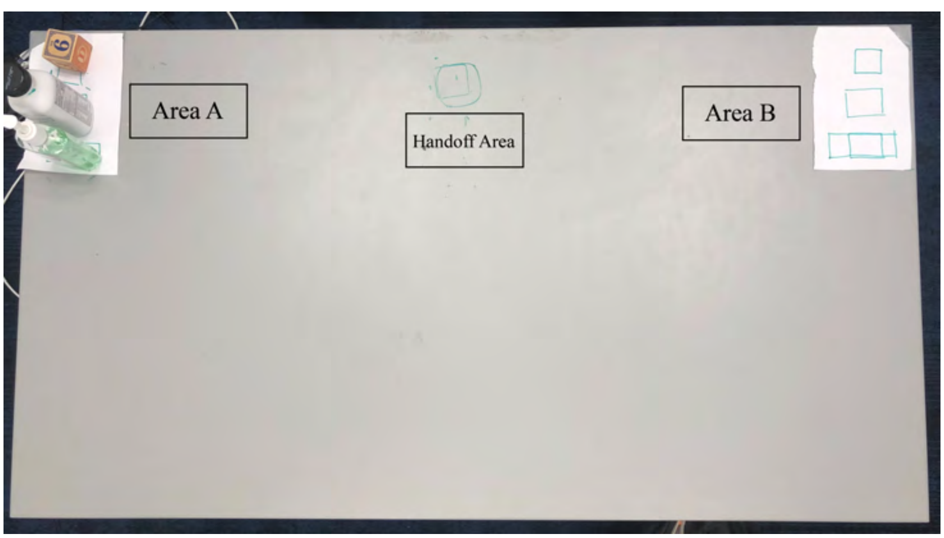

Figure 4. The desk with area labels. In this task, students "taught" Baxter to perform a pick and place of three objects from Area A to Area B, switching limbs at the handoff area.

The pick and place task consisted of teaching Baxter to pick up three differently shaped assigned objects from Area A and placing them on the desk in Area B through both the main mode and demo mode, hence the three videos for each mode. To do so, the students programmed Baxter to perform a handoff motion of the item to the area in the middle, and switch limbs, as Area A and Area B are quite far apart–adding some complexity to this otherwise very simple task. See the area labels in Figure 4.

The three assigned objects are shown in Figure 5, 6, and 7 as follows:

Figure 5. Hand Sanitizer (rectangular cylindrical shape)

Figure 6. Wooden cube (cube shape)

Figure 7. Milk bottle (cubical cylindrical shape)

The main mode will involve the students using ROS Baxter Interface libraries to write a python program. Programming a successful limb path was critical because it ensured the limb was in the correct place to pick-up/drop-off an object and to prevent the limbs from colliding with the table, item, or the other limb.

When moving Baxter's limb from one point in space to another through code, Baxter will always take the shortest path between the two points. Thus, to prevent Baxter from dragging the item along the table (along the x-axis), we must program Baxter to move to two intermediate points: 1) between *A* and *Handoff* on the x-axis and higher than the table in z-axis, and 2) between *A* and *Handoff* on the x-axis and higher than the table in z-axis. We will refer to the first intermediate point as *A'*, and the second as *B'*.

The right limb's path is: A → pick up item (close gripper) → A' → handoff area → place object (open gripper) → neutral position. And then the left limb's path is: handoff area → pick up item (close gripper) →B' → B → place object (open gripper). The right limb must be moved be back to neutral position to avoid colliding with the left limb when it tries to pick up the item from *handoff location*.

A successful limb path can be seen in Figure 8 below:

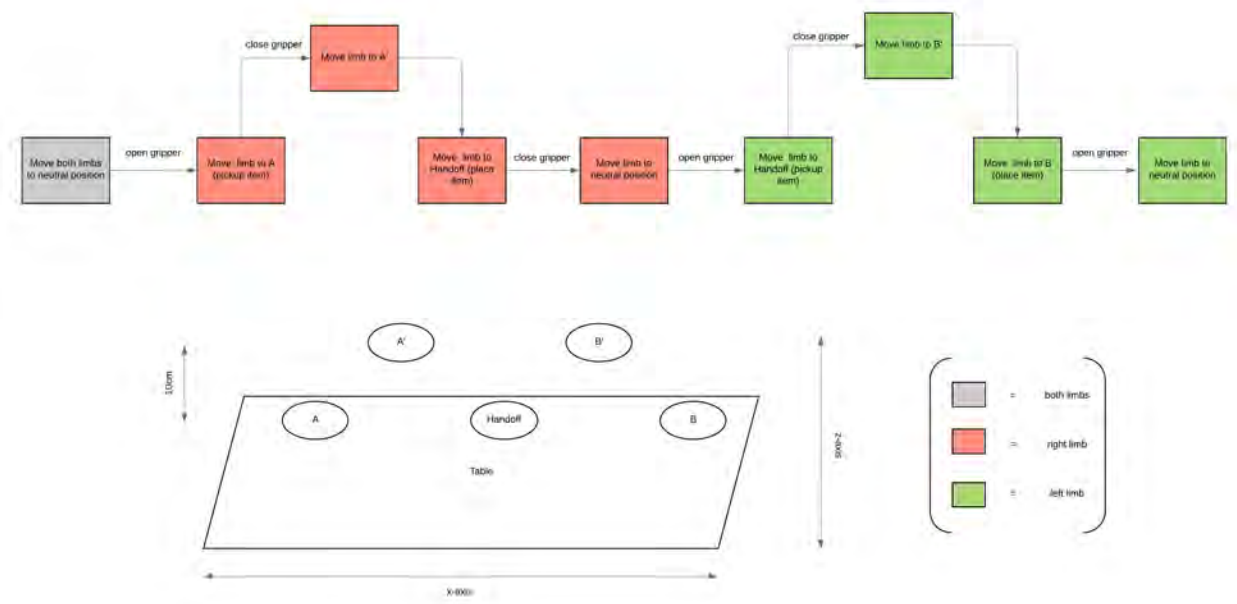

Figure 8. Baxter limb path for picking and placing an item from Area A to Area B with a handoff in between.

An example of the path in code is shown in Figure 9:

```
import sys
import rospy
import baxter_interface

class SimplePickAndPlace(object):
    def __init__(self, limb):
        self._gripper = baxter_interface.Gripper(limb)
        self._gripper.calibrate()
        self._limb = baxter_interface.Limb(limb)
    def move_to_start(self):
        print("move to starting position")
        self._limb.move_to_neutral()
    def pick(self, joint_angles):
        print("picking up object")
        self.move(joint_angles)
        self._gripper.close()
    def place(self, joint_angles):
        print("placing object")
        self.move(joint_angles)
        self._gripper.open()
    def move(self, joint_angles):
        self._limb.move_to_joint_positions(joint_angles)


def main():
    rospy.init_node('simple_pick_and_place')

    right = SimplePickAndPlace('right')
    left = SimplePickAndPlace('left')

    left.move_to_start()
    right.move_to_start()

    pickup_location = {'right_s0': 0.6158932863359936, 'right_s1': -0.499694241653673354, 'right_w0': 0.3094806239558822, 'right_w1': 0.6925923257302643, 'right_w2': -0.444854428
    intermediate_location_right = {'right_s0': 0.9794467330648366, 'right_s1': -0.9284418718676467, 'right_w0': 0.1284788909854034, 'right_w1': 1.3387817326269948, 'right_w2': -
    mid_location_right = {'right_s0': 1.329194352702711, 'right_s1': -0.6181942575178218, 'right_w0': 0.16260196351585385, 'right_w1': 0.3347913069559916, 'right_w2': 0.07056311
    mid_location_left = {'left_w0': 0.059825250727531136, 'left_w1': 1.3004322129298596, 'left_w2': -0.14611167004608566, 'left_e0': 0.1012427320004373, 'left_e1': 1.42506815194
    intermediate_location_left = {'left_w0': -0.20823789195544493, 'left_w1': 0.56105347316909, 'left_w2': -0.5806117282146291, 'left_e0': 0.146495165243057, 'left_e1': 1.556990
    place_location = {'left_w0': 0.23239808936464018, 'left_w1': 0.6665146523362123, 'left_w2': 0.21552430069790063, 'left_e0': -0.012655341500054663, 'left_e1': 1.271286577960

    right.pick(pickup_location)
    right.move(intermediate_location_right)
    right.place(mid_location_right)
    right.move_to_start()
    left.pick(mid_location_left)
    left.move(intermediate_location_left)
    left.place(place_location)

if __name__ == '__main__':
    sys.exit(main())
```

Figure 9. Python code for pick and place in main mode.

The demo mode is more similar to the task creation functionality of the Manufacturing Baxter in that the student will record (or demo) the desired sequence of limb movements and then play it back using the Record/Playback feature, as shown in Figure 10. The limb path is the same between demo mode and main mode. Students can switch between the modes through the Field Service Menu (FSM).

Figure 10. Demo Mode main screen

The final report included a usability evaluation of the function of teaching a Baxter to do a pick and place task for both modes, a comparison of the two modes, and a discussion as to how their findings relate to the recent news that Rethink Robotics has closed its doors and the potential future role of cobots in education. The students were given the freedom to use whatever framework, methodology, and metrics they think is appropriate, but it was suggested that they draw from the theories and frameworks assigned as background reading, particularly the USUS framework due to its holistic approach.

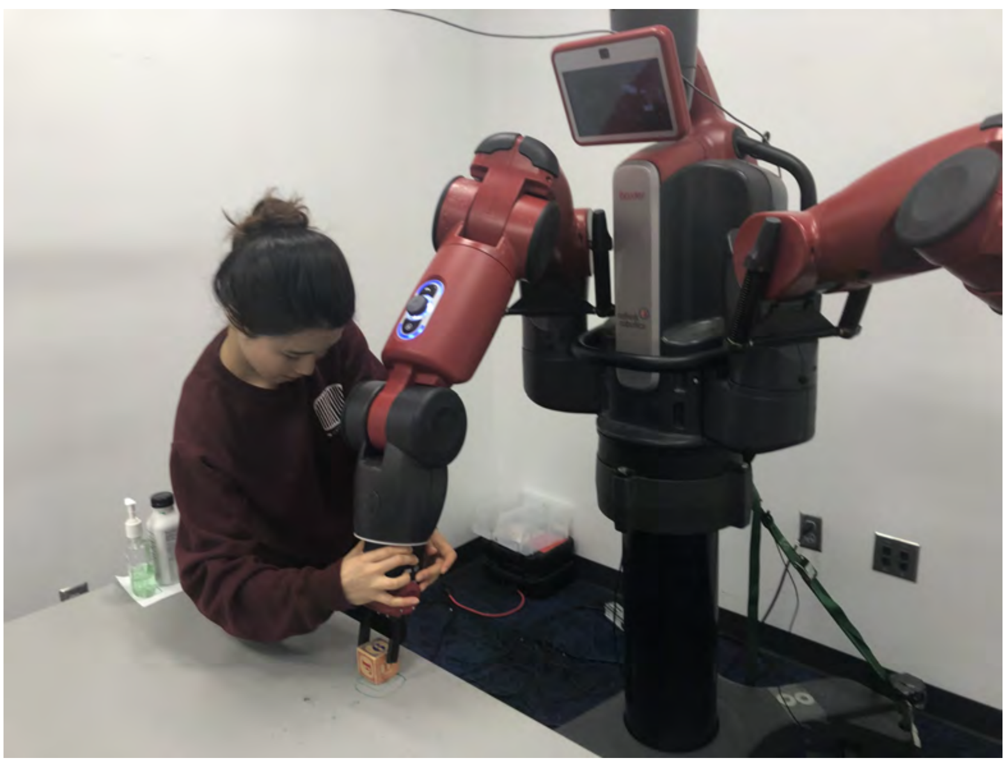

Figure 11. Student guiding Baxter with training cuff. When pressure is put on cuff, the blue Navigator lights illuminates and the arm becomes weightless (zero-gravity mode) and easy to position.

C. Office Hours

For the software installation processes, we held office hours to help students with any technical difficulties. For the parts of the assignment that required direct interaction with Baxter, we asked student pairs to sign up for unlimited 1.5 hour time slots to work with the Baxter robot on the assignment, in which a TA would assist them.

D. Timeline

Although the following due dates were not strictly enforced, it was strongly encouraged that students remain on schedule. The suggested timeline was as follows:

- March 22: Assignment handed out
- March 29: Finish software installation
- April 2: Successfully connect to Baxter
- April 12: Finish pick and place task
- April 16: Assignment due

IV. Results

This section summarizes the results of the project based on student survey and interview responses to gauge the effectiveness of the assignment and Baxter as an educational tool. Five groups chose to interview and four groups chose to do the survey. Questions were the same for both (Appendix B).

Student Usability Test Results

Usability

Most groups felt that the ROS Baxter Python libraries were easy to figure out due to clear documentation, even among those with little prior Python experience. Also all groups agreed the demo mode was straightforward in usability as well.

Baxter had little robustness to correct for errors, which was compounded on top of its lack of precision. An example was when one group accidentally programmed Baxters left limb that was out of its reach. Baxter began to malfunction and needed to be restarted. Baxter is also limited in flexibility as it does not adapt and expected behavior is only possible when given an exactly replicated scenario.

In terms of environment, all groups chose to download a Virtual Machine with Ubuntu on their Mac or Windows computers to run ROS. Nearly all groups ran into issues initially connecting to Baxter and setting up their environment. In addition, due to the variety of hardware and software used by the students, the problems were unique to each individuals machine and caused headaches to the TAs as well. Common problems included difficulty finding the right IP address on their machines, and issues with the ethernet cable.

Overall, the consensus was that the software for both modes supported the basic pick and place task well and were reasonably easy to use but lack of consistency and precision in hardware severely, lack of robustness and flexibility, and system connection and setup difficulties reduced usability.

Social Acceptance

Unsurprisingly, none of the groups ranked Baxter very high on performance expectancy as the effort it takes to teach Baxter such a simple task is much higher than simply doing these tasks by oneself. Most groups felt comfortable with the appearance of Baxter, although one student felt the size and the occasional whirring noises slightly intimidating. Those with less programming experience were initially intimidated by the idea of programming Baxter but felt the process was straightforward once they actually started it.

No groups developed any attachment to Baxter. While I initially anticipated the touch aspect may spark some attachment, most saw it merely as an educational tool. However, some groups did mention referring to Baxter as a he, and speaking encouraging phrases to it, suggesting that although emotional attachment did not develop, students still attributed some human-like characteristics to it.

User Experience

Despite the human-like appearance that Rethink Robotics touted Baxter to have, most students felt that Baxter had very little resemblance to a human. Even though the body shape was modeled after humans (ie. two arms, torso, head), students observed that the way the joints moved, and even its neutral position, was not human-like at all. A couple students felt that teaching Baxter resembled somewhat the process of teaching babies, particularly the part where they need to hold Baxters limbs to demonstrate the exact movement. However, the precision and exactness with which the movement must be demonstrated was decidedly not human-like, as even babies can adapt to adjustments in the environment. Apart from this communication paradigm, students did not feel a strong co-experience with Baxter and did not perceive it to possess any particular personality.

The groups unanimously agreed that they felt very secure working with Baxter, although some mentioned that the speed of the arms (if unadjusted) was intimidating at first.

Societal Impact

About half the students indicated that they were less interested in working with cobots in the future and the other half were more interested after this assignment. And the majority became less optimistic about integrating cobots in society in the future for the purpose of increasing quality of life, health, and security.

Students cited the following as reasons for Rethink Robotics shutdown in 2018: 1) lack of practical use cases, 2) lack of precision, 3) poor user experience, 4) technical knowledge needed to operate the robot, 5) being too general purpose without the ability of doing any specific task well, and 6) the error-prone and time-consuming environment setup.

Students cited the following as future HRI projects they would be interested in working on: 1) computer vision, 2) unmanned aerial vehicles, 3) decision-making models, 4) real-time obstacle avoidance, 5) hardware improvements to increase precision, and 6) creating smoother user experience with cobot interface.

Student Assignment Grades

Out of the nine pairs of students, two received A's, five received B+'s, and two received B's. Based on grades from the previous assignments, results were expected to be higher.

Most common pitfalls were 1) not providing visuals to aid descriptions, 2) not defining objectives of usability testing and why it is important, 3) not defining or inappropriately using quantifiable metrics in usability test, 5) not explaining why they chose the metrics they used, 6) providing insufficient background information, and 7) general lack of clarity and details.

Visuals

Although not required, visuals add clarity to descriptions about Baxter, particularly gripper positions. Diagrams also aid descriptions of processes, and tables represent metrics well.

Usability Testing Objectives

Students should detail objectives of usability testing such as improving construction of robots, and identifying optimal environment and users for the robot evaluated. Students should also identify the main factors of usability tests and describe their importance.

Quantifiable Metrics

Many reports made claims, such as ”Baxter was imprecise”, without any quantifiable metrics for evidence. Other groups used confusing and/or inappropriate metrics. For example, one pair tried to measure effectiveness (a factor of usability in the USUS framework) through ”smoothness level” of picking up each item. This metric is vague and not quantifiable, and offered no comparison between the two modes. Some stronger metrics for effectiveness as a factor of usability would have been success rate and time taken until success for each item and for each mode. Another rated each factor in the USUS Evaluation Framework on a pass/fail basis. This is an improper use of the framework as each factor is meant to be addressed by the methodological mix of quantitative and qualitative measures, not simply rated themselves.

We looked for quantifiable metrics primarily for assessment of the usability factor.

Subjective Metrics

Subjective metrics such as emotions, or degree and/or nature of co-experience with Baxter are valid for many of the factors, but they must be explained with observational details. For example, the claim that ”Baxter is like a child” can be supported by observations about its appearance, sounds, reactions, etc.

Necessary Background Information

Some key points we looked for here was 1) a history of Rethink Robotics as a company and their product lines, 2) overview of Baxter and its main features, and 3) usability testing theories/ frameworks and their objectives.

V. Conclusion

Student feedback indicated some key takeaways: 1) although its safety, and simple ROS Python client libraries were Baxters major upsides, Baxters lack of precision and error-prone environment setup made it an imperfect education tool, and 2) programming is likely not the most effective mode of interacting with a robot for an introductory HRI course project.

Compared against interfacing with Research Edition Baxter through programming, the manufacturing version with the Intera software would have offered a fuller cobot experience through its programming-free task creation functionality, and human-like features (ie. eyes, facial expressions, etc). Although the Baxter Research Editions demo mode mimicked the task creation functionality, the lack of human-oriented perception and features made it hard for students to form attachment to it and perceive it as more than an educational tool. In this way, interacting with a cobot through programming might be more suited for advanced HRI research instead of an introductory HRI course project. However, even if the Manufacturing edition were used, both Baxter editions suffer from lack of precision, leading to student frustration, and possibly even pessimism toward the future of cobots.

In the future, the complex and error-prone environment setup process could be mitigated if a designated machine with all software pre-installed were given to students so they could start working with Baxter right away. Installing all the software from scratch is too time-consuming for a single assignment and should only be done if multiple assignments with Baxter are assigned. Students mentioned that they spent nearly half their time installing software and connecting to Baxter. This is one drawback to Active Learning approach. Although the hands-on aspect allows students to immediately apply concepts learned in lectures on a real robot and platform–merging theory with practice– the technical difficulties in hardware and software that arise can be time-consuming and frustrating. An option that might be worthy to explore is using a simulator instead of a physical robot. Although this will not eliminate problems associated with software, it will eliminate hardware issues (ie. imprecision, etc).

In addition to the explicit goal of teaching the theory and methodology of usability testing to bridge their HRI education with industry applications, an introductory HRI course project should increase student enthusiasm for the future of cobots, or at least provide more insight into their applications in society, both today and speculatively in the future. However, slightly less than half the students indicated that they were more interested in working with cobots in the future and and the remaining became less optimistic about integrating cobots in society in the future, indicating that Baxters lack of precision and error-prone environment set-up process dampened student enthusiasm. Another contributing factor is the portrayal of robots in popular culture Science Fiction as advanced, intelligent, and possibly even superior and poised to take over the human race. The contrast between current robot technology and these fantastical, futuristic depictions may have exacerbated their disappointment.

It is also possible that HRI and working with cobots simply has a steep learning curve. Since most students had never programmed with ROS or interacted with any robot before the course, the dip in enthusiasm could be explained by the lack of experience, and certain ”tribal knowledge” that would have made trivial but tedious tasks, such as setting up the environment, smoother and less time-consuming. The pair with one student who indicated past experience with Baxter finished the project many days before the due date (unlike most pairs), and also received an A (only one of two pairs received an A).

A limitation of this study was the limited sample of students. We had eighteen students total, which formed nine project pairs. More students and reports would have given us more data and feedback on the effectiveness of this project. In conclusion, although its safety, and simple-to-use ROS Python client libraries were Baxters major upsides, its lack of precision, and error-prone environment setup made it an imperfect education tool, and may have decreased student enthusiasm for cobots and HRI. To improve usability testing education in future years of this course project and beyond, we would 1) use a more precise robot or even a simulation program, 2) reserve programming modes of interaction for more advanced courses due to the overhead time needed for environment setup, and 3) emphasize the importance of using quantifiable metrics.

Appendix

Appendix A: Assignment Handout

Baxter Pick and Place Assignment

ME 555.03: Human Robot Interaction

Due: April 16th

Introduction

Growing interest in autonomous humanoid robots in industry has pushed for the development of theoretical and methodological frameworks for Human-Robot collaboration that will improve and standardize the way humans interact with these robots as well as the construction of robots. Important factors that should be addressed in these frameworks include safety, usability, social acceptance, user experience, and societal impact.

To this end, this project will have you test various functionalities of a commonly-used collaborative robot (cobot) called Baxter. When Baxter robots were first introduced, they were heralded as a new form of HRI, where the humanoid robot could learn from demonstration and work alongside humans autonomously.

Your task is to conduct a usability evaluation of the function of “teaching” a Baxter to do a pick and place task through two different modes (programmatically and through demonstration). You will determine which metrics to use, and how to best evaluate what is usability in this context.

What You Will Turn In

- Three videos of Baxter performing task through SDK Main Mode (one for each object)
- The Python file written on main mode
- Three videos of Baxter performing task through SDK Demo Mode (one for each object)
- Final Report (IMRaD format, and one report per team). See the Syllabus for more information.

Part 1: Install Ubuntu + Dual Boot

1. Download the Installer Image file, by picking the ”Desktop CD” image appropriate for your machine:
    - 32-bit (Intel x86): http://releases.ubuntu.com/trusty/ubuntu-14.04.5-desktop-i386.iso
    - 64-bit (AMD64): http://releases.ubuntu.com/trusty/ubuntu-14.04.5-desktop-amd64.iso
2. Create an Ubuntu LiveUSB installer by burning the installer image onto a USB stick.
    - Windows: http://www.ubuntu.com/download/desktop/create-a-usb-stick-on-windows
    - Mac OS X: http://www.ubuntu.com/download/desktop/create-a-usb-stick-on-mac-osx
    - Ubuntu: http://www.ubuntu.com/download/desktop/create-a-usb-stick-on-ubuntu
3. Partition System and Dual Boot Ubuntu
    - Windows: https://www.tecmint.com/install-ubuntu-16-04-alongside-with-windows-10-or-8-in-dual-boot/
    - Mac OS X: https://www.applegazette.com/mac/how-to-install-ubuntu-and-dual-boot-macos/ (start from step 2)

Note: Make sure your partition for Ubuntu is at least 30GB, and make sure the Ubuntu version is 14.04.

Part 2: Install ROS + Set Up Workstation

1. Log in to your Ubuntu system.
2. Follow steps 2 to 7 here: http://sdk.rethinkrobotics.com/wiki/Workstation_Setup

Note: Make sure to install ROS Indigo

Part 3: Hello Baxter!

1. Connect to the router through the Ethernet cable to interact with Baxter.
2. Follow steps here to make sure you are able to connect to Baxter: http://sdk.rethinkrobotics.com/wiki/Hello_Baxter

Part 4: Simple Pick and Place

a) Main Mode

Write a Python program in which Baxter will pick three assigned items respectively from Area A and place items on the desk in Area B. Name the file [netID]_pickandplace.py. Take three separate *videos* of Baxter performing this task for each item.

Note: You must first get the cartesian coordinates of your pick and place locations and convert them to joint positions instead of using the joint positions of these locations directly.

Hints:

1. Think about the trajectory that Baxter will take to pick up and place an object.
2. The following package will be helpful: https://moveit.ros.org/

b) Demo Mode

Another way to "teach" Baxter is through the Demo Mode.

To navigate to this mode, follow these instructions: http://sdk.rethinkrobotics.com/wiki/Demo_Mode

The first screen will have five options. Choose the Record/Playback option. Record the pick and place task you want Baxter to perform and play it back.

As with part 1, take three separate *videos* of Baxter performing the pick and place task from Area A to Area B for each item.

Part 5: Conduct Usability Test For Both Modes

Your final report will be a usability evaluation of the function of "teaching" a Baxter to do a pick and place task for both the SDK Main Mode and SDK Demo Mode. You can use whatever framework you find most appropriate.

Consider using USUS usability testing framework:
https://www.researchgate.net/publication/254200524_The_USUS_evaluation_framework_for_user-centered_HRI

Your final report should compare the methods and include a discussion as to how your findings relate to the recent news that Rethink Robotics, the company who founded Baxter, has closed its doors. Also, discuss the potential future role of cobots in education.

Remember to follow the IMRaD (Introduction, Methods, Result, Discussion) format.

Part 6: Fill Out Survey

Please fill out this survey: http://bit.do/pickandplacesurvey.

Note: One response per person. The deadline of filling out the survey is April 12th.

Appendix B: Survey

Baxter Pick and Place Assignment Survey

ME 555.03: Human Robot Interaction

Due: April 12th

Background

What environment did you have? (ie. virtualbox, dual boot, Mac, etc)

Usability

Note: Interface refers to all the hardware and software components involved (ie. ROS, Baxter limbs, client libraries)

What was the success / task completion rate? (for both modes)

How easy was it to figure out the interface and complete the task once you connected to Baxter? (for both modes)

What was the hardest or least straightforward part of the assignment? (for both modes)

How well was the Baxter suited to support the pick and place task? (for both modes)

Social Acceptance

Were you intimidated at all by Baxter or the assignment in general? If so, by what part? (ie. programming, working with a cobot, the appearance of Baxter, etc)

Did you form any emotional attachment to Baxter while completing the assignment during either mode? If so, in what way?

User Experience

What were the primary emotions you felt as you were completing this task?

What humanistic traits of Baxter stuck out to you the most?

Did you feel secure while working with Baxter?

To what extent was your interaction and communication with Baxter similar to that with a human?

Societal Impact

What applications do you see Baxter in improving people's' quality of life, health, and security?

Now that you have some familiarity with the Baxter interface, describe what kind of HRI project you would be interested in working on next.

Would you work with Baxter again?

Why do you think Rethink Robotics shut down in 2018?

Did this assignment and working with Baxter make you more or less interested in working with cobots in the future in your education?

Did this assignment and working with Baxter make you more or less optimistic about integrating cobots in society in the future?

Assignment

Did you feel that you were given enough time to complete the assignment?

References

[1] Goodrich, M. A., Schultz, A. C., et al., "Human–Robot Interaction: A Survey," *Foundations and Trends® in Human–Computer Interaction*, Vol. 1, No. 3, 2008, pp. 203–275.

[2] Fong, T., Thorpe, C., and Baur, C., "Collaboration, Dialogue, Human-Robot Interaction," *Robotics Research*, Springer, 2003, pp. 255–266.

[3] Chen, J. Y. and Barnes, M. J., "Human–Agent Teaming for Multirobot Control: A review of Human Factors Issues," *IEEE Transactions on Human-Machine Systems*, Vol. 44, No. 1, 2014, pp. 13–29.

[4] Sheridan, T. B., "Human–Robot Interaction: Status and Challenges," *Human Factors*, Vol. 58, No. 4, 2016, pp. 525–532.

[5] Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A., and Dong, J.-J., "A Review of The Applicability of Robots in Education," *Journal of Technology in Education and Learning*, Vol. 1, No. 209-0015, 2013, pp. 13.

[6] Causo, A., Vo, G. T., Chen, I.-M., and Yeo, S. H., "Design of robots used as education companion and tutor," *Robotics and Mechatronics*, Springer, 2016, pp. 75–84.

[7] Weiss, A., Bernhaupt, R., and Tscheligi, M., "The USUS Evaluation Framework for User-Centered HRI," *New Frontiers in Human–Robot Interaction*, Vol. 2, 2011, pp. 89–110.

[8] Dumas, J. S., Dumas, J. S., and Redish, J., *A Practical Guide to Usability Testing*, Intellect Books, 1999.

[9] Zhang, D. and Adipat, B., "Challenges, Methodologies, and Issues in The Usability Testing of Mobile Applications," *International Journal of Human-Computer Interaction*, Vol. 18, No. 3, 2005, pp. 293–308.

[10] Bastien, J. C., "Usability Testing: A Review of Some Methodological and Technical Aspects of The Method," *International Journal of Medical Informatics*, Vol. 79, No. 4, 2010, pp. e18–e23.

[11] Endo, Y., MacKenzie, D. C., and Arkin, R. C., "Usability Evaluation of High-Level User Assistance for Robot Mission Specification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 34, No. 2, 2004, pp. 168–180.

[12] Yanco, H. A., Drury, J. L., and Scholtz, J., "Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition," *Human-Computer Interaction*, Vol. 19, No. 1, 2004, pp. 117–149.

[13] Brooke, J., "SUS-A Quick and Dirty Usability Scale," *Usability Evaluation in Industry*, Vol. 189, No. 194, 1996, pp. 4–7.

[14] Cummings, M. L. and Donmez, B., *Metrics for Supervisory Control System Evaluation*, London, UK: Oxford Univ. Press, 2013.

[15] Riek, L. D., "Embodied Computation: An Active-Learning Approach to Mobile Robotics Education," *IEEE Transactions on Education*, Vol. 56, No. 1, 2013, pp. 67–72.

[16] "Baxter Overview," http://sdk.rethinkrobotics.com/wiki/Baxter_Overview, 2015, Accessed: 2019-04.

[17] "About ROS," http://www.ros.org/about-ros/, 2019, Accessed: 2019-04.

[18] Fitzgerald, C., "Developing Baxter," *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*, IEEE, 2013, pp. 1–6.

[19] "First Task," http://mfg.rethinkrobotics.com/wiki/First_Task, 2015, Accessed: 2019-04.