

University of Louisville

ThinkIR: The University of Louisville's Institutional Repository

Electronic Theses and Dissertations

8-2017

An interactive interface for nursing robots.

Ankita Sahu
University of Louisville

Follow this and additional works at: <https://ir.library.louisville.edu/etd>



Part of the [Computer and Systems Architecture Commons](#), and the [Robotics Commons](#)

Recommended Citation

Sahu, Ankita, "An interactive interface for nursing robots." (2017). *Electronic Theses and Dissertations*. Paper 2796.
<https://doi.org/10.18297/etd/2796>

This Master's Thesis is brought to you for free and open access by ThinkIR: The University of Louisville's Institutional Repository. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of ThinkIR: The University of Louisville's Institutional Repository. This title appears here courtesy of the author, who has retained all other copyrights. For more information, please contact thinkir@louisville.edu.

AN INTERACTIVE INTERFACE FOR NURSING ROBOTS

By

Ankita Sahu
B.Tech, Department of Biotechnology
Visvesvaraya Technological University, 2010

A Thesis
Submitted to the Faculty of the
J.B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science
in Computer Science

Department of Computer Engineering and Computer Science
University of Louisville
Louisville, Kentucky

August 2017

Copyright 2017 by Ankita Sahu

All rights reserved

AN INTERACTIVE INTERFACE FOR NURSING ROBOTS

By

Ankita Sahu
B.Tech, Department of Biotechnology
Visvesvaraya Technological University, 2010

A Thesis Approved On

4th August 2017

by the following Thesis Committee:

Olfa Nasraoui, Ph.D.

Dan O. Popa, Ph.D.

Nihat Altiparmak, Ph.D.

ACKNOWLEDGEMENTS

First and foremost I thank Dr. Dan O. Popa and Dr. Olfa Nasraoui for their constant support and guidance throughout my research work and Masters studies. I am thankful for all the opportunities and resources that were available to me at the Next Generation Systems(NGS) lab. Working at NGS has given me a broad exposure and a unique perspective into the world of robotics. I really enjoyed working with different kinds of robot and building my skills during my research work. I would like to thank Dr. Nasraoui for advising me and always being there to guide irrespective of the time of day. Her constant motivation helped me move forward with my research work. Our discussion on the speech interface was enlightening and informative. I would also like to thank Dr. Nihat Altiparmak for taking the time out of his busy schedule and serving on my thesis committee

I would like to especially thank Dr. Indika Wijayasinghe and Dr. Sven Cremer for mentoring me during the course of my research work. I want to thank my colleagues at NGS - Ruoshi Zhang, Abubakar Shamsudeen, Joshua Baptist, Nasser Saadatzi, Chris Robinson, Danming Wei, Zhong Yang, Brandon Young, Chris Newquist and Yathartha Tuladhar. It was a memorable experience working with you guys and thank you for helping me out whenever I needed it.

I would like to thank my parents Ram Ratan Sahu and Saraswati Sahu for their unconditional love, support, and sacrifices without none of this would have been possible. Their constant motivation to do better than what I am has been the driving force of my life. I would also like to thank my brother Ashwani Sahu and my sister-in-law Bimla Bhatt for their love and support. Thank you for believing in me and encouraging me to pursue my dreams. I am thankful to my husband, Sumit Kumar Das for having patience and trying

to get the best out of me. You mentored me like a teacher and supported me like a friend.
I would not have achieved this without your support and constant tantrums to work hard.

ABSTRACT

AN INTERACTIVE INTERFACE FOR NURSING ROBOTS

Ankita Sahu

August 4, 2017

Physical Human-Robot Interaction (pHRI) is inevitable for a human user while working with assistive robots. There are various aspects of pHRI, such as choosing the interface, type of control schemes implemented and the modes of interaction. The research work presented in this thesis concentrates on a health-care assistive robot called Adaptive Robot Nursing Assistant (ARNA). An assistive robot in a health-care environment has to be able to perform routine tasks and be aware of the surrounding environment at the same time. In order to operate the robot, a teleoperation based interaction would be tedious for some patients as it would require a high level of concentration and can cause cognitive fatigue. It would also require a learning curve for the user in order to teleoperate the robot efficiently. The research work involves the development of a proposed Human-Machine Interface (HMI) framework which integrates the decision-making module, interaction module, and a tablet interface module. The HMI framework integrates a traded control based interaction which allows the robot to take decisions on planning and executing a task while the user only has to specify the task through a tablet interface. According to the preliminary experiments conducted as a part of this thesis, the traded control based approach allows a novice user to operate the robot with the same efficiency as an expert user.

Past researchers have shown that during a conversation with a speech interface, a user would feel disengaged if the answers received from the interface are not in the context of

the conversation. The research work in this thesis explores the different possibilities of implementing a speech interface that would be able to reply to any conversational queries from the user. A speech interface was developed by creating a semantic space out of Wikipedia database using Latent Semantic Analysis (LSA). This allowed the speech interface to have a wide knowledge-base and be able to maintain a conversation in the same context as intended by the user. This interface was developed as a web-service and was deployed on two different robots to exhibit its portability and the ease of implementation with any other robot.

In the work presented, a tablet application was developed which integrates speech interface and an onscreen button interface to execute tasks through ARNA robot. This tablet interface application can access video feed and sensor data from robots, assist the user with decision making during pick and place operations, monitor the user health over time, and provide conversational dialogue during sitting sessions. In this thesis, we present the software and hardware framework that enable a patient sitter HMI, and together with experimental results with a small number of users that demonstrate that the concept is sound and scalable.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
ABSTRACT	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Speech Interface for Conversational Abilities	6
1.3 Research Contributions of this Thesis	7
1.3.1 Human-Machine Interface (HMI) Framework	7
1.3.2 Speech Interface Knowledge-base	8
1.3.3 Code and Papers Published	8
1.4 Thesis Organization	9
2 LITERATURE REVIEW	10
2.1 Assistive Robots	10
2.2 Tablet Interface	13
2.3 Speech Interface	14
3 SYSTEM DESCRIPTION	16
3.1 PKD Hardware Description	16
3.2 Baxter Robot	18
3.3 ROS	20
3.4 Microsoft Kinect [®]	20
3.5 Google Pixel C Tablet	22

4	HUMAN MACHINE INTERFACE FRAMEWORK	25
4.1	System Architecture	28
4.2	Integrating Tablet and ROS	31
5	SPEECH INTERFACE	34
5.1	Web service	34
5.2	AIML Knowledge Base	37
5.3	Wikipedia Knowledge Base	40
5.3.1	Latent Semantic Analysis(LSA)	41
6	EXPERIMENTS AND RESULTS	43
6.1	Human Machine Interface	43
6.2	Speech Engine	46
7	CONCLUSION AND FUTURE WORK	55
7.1	Conclusion	55
7.1.1	Human-Machine Interface (HMI) Framework	55
7.1.2	Tablet Interface	56
7.1.3	Speech Interface	57
7.2	Future Work	58
	REFERENCES	59
	CURRICULUM VITAE	65

LIST OF TABLES

1	AIML tags description	37
2	Average rating of the interface by the users	46
3	Easy Questions	47
4	Medium Questions	49
5	Hard Questions	51
6	LSA Result	53

LIST OF FIGURES

1	Control framework between user and robot [1]	4
2	Software Framework for ARNA Sitter Robot	5
3	From Left to Right: Flo robot for elderly care [2], HERB: Home Exploring Robot Butler [3]	11
4	From Left to Right: Care-O-Bot 3 [4], RI-MAN [5], Wakamaru [6]	12
5	PKD in the NGS lab	17
6	Baxter Robot	19
7	Baxter Gripper	19
8	Microsoft Kinect Components [7,8]	21
9	Kinect Camera on Baxter Head	22
10	Google Pixel C Tablet	23
11	User Interface on Google Pixel C Tablet	24
12	User Interface for interacting with PKD	24
13	Hospital room and patient bed [9]	26
14	System Overview	28
15	Proposed Human Machine Interface Framework	29
16	SOAP message skeleton	35
17	SOAP Web service HTTP Protocol	36
18	Web service Flowchart	36
19	Example of AIML	40
20	Example of <srail>tag in AIML	40
21	Markers for object identification	43
22	Time of completion comparison between users	45

CHAPTER 1

INTRODUCTION

1.1 Motivation

Recently, robots are increasingly being used in social contexts while assisting human users in daily activities. In general, assistive robots can be defined as robots which can perform activities to benefit users who may or may not have disabilities. David et. al. has defined several aspects of socially assistive robots based on different properties of the robot such as the user population, required tasks to be performed etc. [10].

There have been eminent developments of such assistive robots intended to help elderly people and medical staff in a hospital environment. Studies on deploying robots in healthcare facilities have shown them to be promising in improving the recovery of patients in physical therapy exercises [11]. Use of these kinds of robots in a healthcare facility can also lead to a reduction of routine tasks and workload on nurses. Assistive robotics have been observed to help elderly users in a household environment as well [3]. In order to perform their tasks efficiently, assistive robots generally contain an array of onboard sensors to help the system sense obstacles, plan motion trajectories and carry out instructed tasks. Many of these sensors are used routinely on robots deployed in factories and warehouses.

When working in a healthcare environment with sick individuals, assistive robots need to be particularly safe during assistance rendered to patients, and therefore, additional soft covers and sensors such as robotic skin for measuring physical pressures exerted onto patients are needed. During assistance, the sharing of work spaces must be done safely such that the users are not hurt by the robot. Studies have been conducted at defining the safety features to be included in assistive robots [12–14]. The studies conducted by Virk et. al. provides an overview of the ISO TC184/SC2 standard [14]. These standards and

safety precautions are necessary as they ensure the safety of the user and account for all foreseeable contingencies.

An assistive robot is classified as a “medical robot” if it is intended for personal care and monitoring of a user and may or may not conduct invasive procedures to improve the quality of health [14]. These types of robots need to be able to detect physical human interaction and avert the hazards associated with high force contact [14]. Such capabilities would require the robot to slow down its motion and have a high-frequency control loop running to detect and react to physical contact with the user. There should also be an ease of use factor associated with assistive robots to reduce the learning curve for their use. If the robot is easy to use through an intuitive interface, the user will also be able to harness its full capabilities.

In a healthcare facility, the use of a robot to perform routine tasks can be broadly divided into two areas: walker robots and sitter robots [9]. The Adaptive Robot Nursing Assistants (ARNA), as identified by Cremer et. al. [9], have their tasks divided based on their functional nature [9]. An ARNA walker robot assists patients while walking by providing partial support and gait stabilization according to the individual health condition of patients. During walking, the robot has to carry all necessary medical equipment needed by the patient. This requires the robot to be sturdy enough to provide the necessary support without being unstable due to the applied loads. This also entails the robot to match its “walking speed” to that of the patient so that it is neither running off from the patient nor is hindering the patient’s gait.

An ARNA sitter robot, on the other hand, is required to perform tasks similar to its human counterpart known as a patient sitter. The robot has to monitor the patient and raise an alarm to the attending nurse if the patient is about to fall off or the patient is getting off the bed against medical advice. The robot is also required to warn the patient of the danger of getting out of the bed while alerting the nurse. The ARNA sitter robot also has to monitor and record the patient’s vitals at regular intervals which will be accessible to caregivers and nurses. If any anomalies are found during these checks, the nurses will

be alerted. Apart from the medical functions, the robot has to be social with the patient as well. It should fetch items as directed by the patient and strike a conversation with the patient. These activities require mobile manipulation capabilities and a knowledge base for a Chatbot based system. To fetch items as requested by the patient, the robot is required to do path planning, obstacle avoidance, and grasp strategy planning. All these activities should be planned and executed keeping in mind the small space available in the room. For having a conversation with the patient, the robot needs to have a knowledge repository from which it will be able to gather answers or responses to the patient. The interactions between the user and the robot should be completed through a human-robot interface or HMI.

A human-robot interface or HMI facilitates a smooth interaction between a user and a robot. These interfaces can be custom made to a specific purpose or can be as generic as control through mobile applications on a tablet. Using tablet interfaces to control robots is not uncommon and is currently gaining more popularity as tablets are easily accessible. Tablet applications can be easily developed and deployed on the devices to control the robots. Studies have suggested that in order to teleoperate a robot, there is a distinct difference between the performance of a novice and an expert user [15,16]. Differences arise due to the amount of practice and the learning curve involved in controlling the robot. This also gives the human user the power for all the decision making on behalf of the robot, while the robot simply follows commands from the user. However, such an interface is not suitable for users who are patients at a hospital instead an easier interface should be designed to allow the user to easily command the robot to perform certain tasks autonomously rather than continuously teleoperating the robot.

In past work, researchers have put forth different types of controls schemes in a user-robot interaction framework. The different components of the controls between the user and the robot can be seen in Figure 1. The communication channel in the framework is responsible for delivering messages and commands back and forth between the user and the robot. The different types of control schemes in this framework are traded control

and shared control [1]. In a shared control, continuous monitoring of the robot is required during the process of completion of tasks. This requires constant attention from the user throughout the robot activity. Traded control, on the other hand, requires the user to initiate a task and the planning and execution of the task is carried out by the robot itself. This type of control requires less attention from the user as well as reduces the prolonged need of holding the interface device throughout the tasks. It also reduces the learning curve required to operate the robot. In this thesis, we present the work that has implemented this kind of control on the ARNA sitter robot to facilitate an easy interface between the patients and the robot.

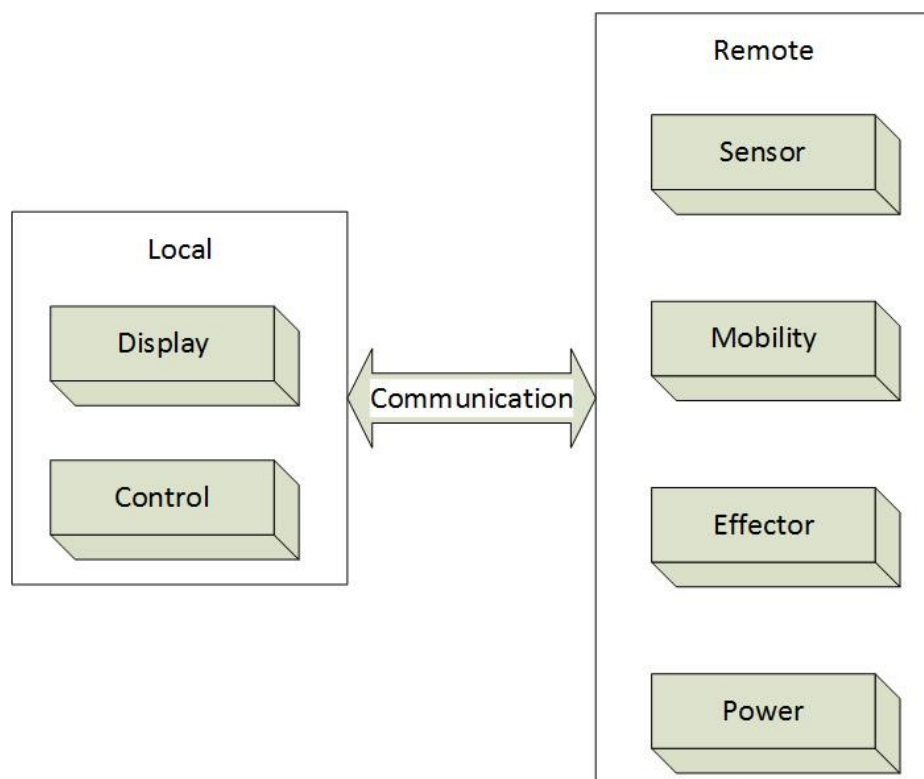


Figure 1: Control framework between user and robot [1]

A software framework designed for a robot system ensures that all the functionalities are in place and are triggered by appropriate stimulus. In the work presented in this thesis, a software framework was developed to accommodate all the functionalities required by the ARNA sitter robot. Figure 2 shows the framework implemented on the ARNA sitter

robot. The framework accommodates user to input both in the form of verbal commands as well as on screen button commands of the tablet interface. The framework lays down the foundation of integrating the user and the robot through a human-robot interface device, which is a hand-held tablet in this case.

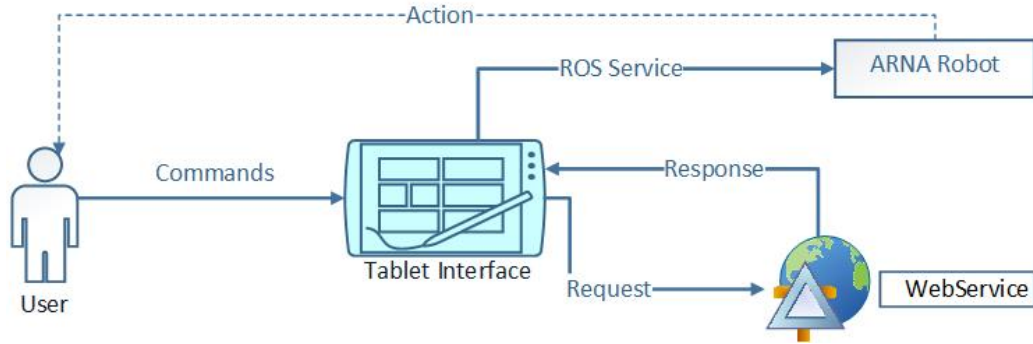


Figure 2: Software Framework for ARNA Sitter Robot

In the proposed framework, a web service server is introduced to run the speech interface engine. This web service server can accept requests that have been translated from verbal commands to text, and send an appropriate response relevant to the conversation. The response is then converted to speech through a text-to-speech engine. The web service is also capable of differentiating between a conversation and a command for the robot. If the command is issued by the user, the web service directs the robot to perform the respective task.

ROS (Robot Operating System) nodes for the robot system are also integrated into the proposed framework. ROS is an emerging standard for robots and is explained in section 3.3. The ROS nodes are also able to communicate with the web service server to perform tasks initiated through verbal commands. This type of framework utilizes the traded control approach which helps a patient to instruct the robot to perform certain tasks rather than having to monitor the robot continuously. The patient can talk to the robot and switch to verbal commands seamlessly with the help of the web service server integrated into the framework.

1.2 Speech Interface for Conversational Abilities

The communication between a human being and machine has been made possible by using natural language processing algorithms used in artificially intelligent chatbots. The means of communication can be auditory or textual mode based on the bot. The development of chatbots started in 1966 with the chatbot named ELIZA followed by PARRY in 1972 and since then, there have been a plethora of chatbots developed to serve distinct objectives [17]. All these systems utilize knowledge bases to serve their objective.

The knowledge base plays a vital role in making a chatbot intelligent. It helps in transforming queries from the user to a structured response. The most common knowledge base used is the Artificial Intelligence Markup Language (AIML) which consists of a set of patterns of question and answer. The AIML is the most widely used knowledge base as it can be modified according to the requirement of the chatbot such as Donna, a personal assistant [18], Dorothy, a network Management chatbot [19], a multimodal virtual guide [20], Charlie, an educational interface bot [21] and TutorBot, an e-learning platform [22]. Among other knowledge bases are online forums, unstructured documents and semi-structured data [23, 24]. For instance, one source of knowledge is Wikipedia, which is a semi-structured database with more than 5,437,095 articles in English version only [25]. It is an online encyclopedia available for free. Since it contains information on a vast area of topics, it is helpful to be used as a knowledge base.

The most important feature of a chatbot is to have the conversation relevant to the topic the user is talking about. If the response of the chatbot is not relevant to the topic, the user may become disinterested. Hence, we have used the AIML knowledge base along with another knowledge base created using Wikipedia documents. The second knowledge base uses Latent Semantic Analysis (LSA) [26] to create a semantic space containing projections of Wikipedia documents so that the responses from the chatbot are relevant to the topic. Hence, the semantic space that generates a relationship between documents helps in making sure the response is related to the topic.

1.3 Research Contributions of this Thesis

The research work in this thesis is aimed at building a framework for the ARNA sitter robot functionalities. This framework is not robot dependent and can be used on any robot meant to serve as an ARNA sitter. Another research contribution of the work presented is the development of the chatbot engine. The chatbot developed is able to maintain a relevant conversation with the user. This was achieved by building a broader knowledge base for the chatbot using Wikipedia's information. These developments are meant to be integrated with the ARNA sitter robot.

1.3.1 Human-Machine Interface (HMI) Framework

The HMI framework introduced in this thesis was developed taking into consideration that the framework would be implemented on robots that are deployed in a hospital environment. This requires less input from users, emergency stop in case of contingencies and putting in place some rules to ensure overall safety. The framework uses traded control which delegates the planning and execution of tasks to the robot and only takes the description of tasks as the input from the user. This type of control is viable to be used by a patient in the hospital.

The framework also implements an emergency stop through the tablet application. This helps in shutting down the robot by the user at any stage of task execution. The emergency stop may be used when the execution of tasks is not progressing in a manner that is acceptable to the user. The framework also implements the use of verbal commands as well as on screen button commands and they are inter-operable. This gives the user the flexibility to switch between the input modes. The verbal command based emergency stop also ensures that in case the user is not able to hold the tablet for long, the stop command is still executed through voice. This also ensures the overall safety features of the robot.

The framework was evaluated using a number of experiments with lab users. Both expert and novice users were asked to perform object fetch tasks through the tablet mobile app. Results, including completion times, failure rates, and overall user satisfaction with the

interface was recorded and form the basis for the recommendation in future implementations on the ARNA robot.

1.3.2 Speech Interface Knowledge-base

In order to keep a user engaged and interested in a conversation with a robot using a chatting interface, the chatbot has to reply within the context of the conversation being held [27]. Therefore, in this project, we developed a knowledge base for the ARNA chatbot. This knowledge base uses Wikipedia and AIML (Artificial Intelligence Markup Language) as the repository. The database was built by using the files from Wikipedia and generating a semantic space using LSA (Latent Semantic Analysis) to find a relation between the documents and the query. We have also experimented with adding answers from public forums to the knowledge base.

During the course of interaction, the generated knowledge base will provide context to the robot while conversing with a user. This feature is particularly helpful for a patient sitter robot as the patient could talk to the robot. The answers from unrelated topics are also removed during this conversation, thus keeping the patient engaged. The chatbot was evaluated experimentally in the lab with a small number of users, and the results are reported in this thesis.

1.3.3 Code and Papers Published

There were open source ROS packages developed during the course of the work presented here. The ROS packages were implemented based on the framework described earlier and the ROS nodes pertaining to the framework can be easily used on any robot for the purpose.

The packages can be found at:

“<https://bitbucket.org/account/user/nextgensystems/projects/TIAS>”.

In addition, the research work into the ARNA sitter robot framework was published at SPIE 2017,

Sumit Kumar Das; Ankita Sahu; Dan O. Popa, “Mobile app for human-interaction

with sitter robots,” in Proc. SPIE. 10216, Smart Biomedical and Physiological Sensor Technology XIV, 102160D. (May 16, 2017) doi: 10.1117/12.2262792.

1.4 Thesis Organization

Chapter 2 contains a literature review about assistive robots, tablet interface and speech interface. In our review of assistive robots, we present a description of different types of robots available or developed for catering in household and health-care environments. In our review of tablet interfaces, we describe existing work done to integrate ROS with Android applications. Finally, in our review of speech interfaces, we discuss the chatbot engines and existing knowledge bases that have been developed.

Chapter 3 explains, in detail, the hardware used for the experiments conducted for this thesis. The descriptions include the android head, Philip K. Dick (PKD), Baxter robot and other hardware such as the Google Pixel C tablet for user interface, Asus Xtion Pro camera for object tracking and Microsoft Kinect camera for the robot view.

Chapter 4 describes the software framework of the ARNA robot. A detailed explanation is presented of what has been implemented in the Android application deployed on the Google Pixel C tablet and the modules created. This chapter also includes an explanation of the Cortex module developed for the ARNA robot.

Chapter 5 explains the chatbot interface developed and implemented in Java. It contains a detailed explanation of the web service and the knowledge base in AIML (Artificial Intelligence Markup Language) and the LSA based approach used to create a semantic space using Wikipedia documents.

Chapter 6 contains information about the experimental setup and the results. The experimental setup includes the objects used in the experimentation and all the steps involved in conducting the experiments. It also includes a discussion of the results obtained from those experiments.

Finally, Chapter 7 summarizes the contributions made in this thesis, and finishes with our conclusions and future work.

CHAPTER 2

LITERATURE REVIEW

Assistive robots are designed to help human users to accomplish certain tasks. Assistive robots can be classified into different categories depending on their area of implementation and the type of assistance rendered. Some of the classification categories presented by David and Maja are user population-based, task-based, and interaction-based [10].

Along with developing assistive robotics, the development of interfaces to interact with the robots is also necessary. These interfaces are able to interpret the human intention or motive and direct the robot to perform the tasks accordingly.

In this section, we discuss past work in the development and classification of assistive robotics and various human-robot interfaces.

2.1 Assistive Robots

An assistive robot can be classified as assistive elderly care robot, assistive prosthetic robot, assistive education robots, etc., based on the section of human population it caters to [10]. Based on the tasks performed, the robots can also be classified as therapy robots, life assistant robots, emotional and expressive robots, etc. [10]. Another classification basis is the interaction interfaces required to communicate with the robot, such as speech, gesture, touch, etc. [10].

A robot named Flo, shown in Figure 3, was created by Nicholas et al. to assist elderly people in a household environment [2]. It is capable of safeguarding the users from physical injury, reminding the users of important tasks such as taking medicines on time, collecting diagnostics data continuously, and social interaction with the user. It also has the capability of remote telemedicine, where the doctor from the clinic will be able to examine the patient at home. Another such robot developed to work in the household environment

is HERB or Home Exploring Robot Butler [3]. HERB was developed to carry out tasks and explore the household environment seamlessly. The idea of the development of HERB was to develop a reliable robot to complete assigned tasks at a similar speed to a human user so that the user is not frustrated using the robot [3].



(a)



(b)

Figure 3: From Left to Right: Flo robot for elderly care [2], HERB: Home Exploring Robot Butler [3]

Assistive robots deployed in a healthcare environment will be able to perform the tasks of a caregiver by accomplishing daily routine tasks such as diagnostics and data collection. They can also perform conversations with the patients to keep them company. The human factors affecting the acceptance of healthcare assistive robot have been laid out by Broadbent et al. [28]. Some of these factors include age, needs, experience with technology, cognitive ability, anxiety and attitudes towards robots etc. [28]. Broadbent et. al. also laid out the robot factors affecting the acceptance of healthcare robots which include appearance, humanness, size, adaptability etc. [28]. Keeping these factors in sight the assistive robots can be developed to be used as assistants to nurses. Some of the assistive robots developed to be deployed alongside human users for assistance are Care-O-Bot [4],

RI-MAN [5]. and Wakamaru [6] as seen in Figure 4.



(a)

(b)



(c)

Figure 4: From Left to Right: Care-O-Bot 3 [4], RI-MAN [5], Wakamaru [6]

Applications of assistive robots in healthcare, more specifically in nursing assistant and home care, have been recognized as important tools to aid human activity. Cremer and Doelling [9] describe a typical operation of a sitter robot in a hospital environment, that can carry out conversational interactions with a non-ambulatory patient, fetch objects from the environment, monitor patient vital signs, and inform the nurse if abnormal patient conditions have been detected. In a health-care facility setting, these conditions need to be met to ensure safety. The HMI framework must assist the robot in implementing software-based safety guidelines to make it usable by a patient or a nurse.

There have been several studies on the development of interfaces to help the human user interact with the robot with ease. For health-care based assistive robots, the design of the Human-Machine interface should be simple and easy to use for the patient operating the robot.

2.2 Tablet Interface

Mobile phones and tablets have become pervasive and contain powerful processors onboard to perform complex tasks. The use of smartphones to control mobile robots has been recently investigated [29, 30]. Mobile applications can communicate with the robots over a network and process data received from the robot. As discussed in several theses [29–31], Android applications can be developed to harness the onboard sensors of the smartphone or the tablet [32].

Alonzo et al. [16] and Cremer et al. [15] have exhibited the use of android based tablets to teleoperate and control robots. In order to integrate ROS with the tablet, rosjava [33] was used. Rosjava provides libraries that can be used to communicate with other devices running ROS and allows the data exchange necessary for the ROS ecosystem. Rosjava and an android tablet can thus be used to control robots.

Utilizing a tablet as a user interface provides an ease of use to the user controlling the robot. Using the tablet for teleoperating a robot has performance differences between an expert user and a novice user as explained by Cremer [15]. Teleoperation requires constant attention from the user as well as expertise in using the interface. This involves teaching new users the interface and practice to control the robot. Another scheme of controlling the robot would be to initiate tasks from the tablet. The required tasks can be planned and executed autonomously. This would require no training on the interface for a novice user. In all these cases, tablet interfaces can be used as instruments to control the robot.

2.3 Speech Interface

The speech interface allows the user to interact with the assistive robot using spoken words. These can be either for conversational purposes or issuing a verbal command to the robot. This type of interface discounts the need of holding a controller and is relevant to the scenario of healthcare based assistive robots.

For conversational abilities, a chatbot program has to be installed on the robot. The most common chatbot program is based on AIML or Artificial Intelligence Markup Language [34]. AIML contains the knowledge base of the chatbot. AIML can also describe the behavior of the chatbot. ALICE or Artificial Linguistic Internet Computer Entity chatbot was developed based on AIML files [34]

In order to impart conversational abilities to the robots, they must be equipped with the knowledge of the subject being spoken about and if the knowledge base is not present in the database, the chatbot should be able to search the internet for answers pertaining to specific topics. In the research work published by Shumaker et. al. [27], it was observed that users were more interested when they had a conversation with a domain knowledge chatbot rather than just a generic chatbot conducting a normal conversation. To include domain specific knowledge to the chatbot, domain specific AIML files were generated and included into the framework.

Along with speech, the emotion or expressiveness of the chatbot is also important to exude a sense of familiarity during the conversations. In research presented by Vogt et al. [35], the chatbot is able to sense the emotions of the speaker based on the pitch, intensity, speaking rate and voice quality of the words spoken. These parameters help the chatbot decide the emotional state of the user and can be used to formulate an appropriate response. Another such research focuses on imparting humor to a chatbot [36]. In this work, the ALICE framework is embedded into a Yahoo Messenger client. It generates face expressions according to the humoristic context of the text. Recognition of the humoristic text is done using the methods below:

- CMU dictionary- used for phonetic transcription of the sentence. The repetition of

the initial phoneme helps in finding the alliteration.

- WordNet - used for finding the semantic relation between nouns, verbs, adverbs and adjectives to build an antinomy relation between them.

A semantic layer development for an intuitive chatbot was presented by Agnese et. al. [24]. It uses both AIML and DBpedia as the knowledge base to retrieve answers. DBpedia is a knowledge base which extracts structured information from Wikipedia. Data can be retrieved using the SPARQL Protocol and RDF (Resource Description Framework) Query language (SPARQL). It uses Latent Semantic Analysis (LSA) to project the documents into a new space where the geometric similarity captures the semantic relationship between different documents. This new space helps the chatbot in formulating its response regarding a specific discussion topic.

Chatbots were also developed to work as personal assistants [18]. The personal assistant chatbot is web-based. Extraction of data is done using the Stanford Core NLP(Natural Language Processing) Libraries and DBpedia. The Stanford CoreNLP libraries are a set of natural language analysis tools which take text input and perform lemmatization, POS(part-of-speech) tagging, and markup the structure of sentences in terms of phrases and word dependencies. Designing the conversation is done using an open-source software called ChatScript. ChatScript is a scripting language which accepts user text input and generates a text response.

The latent semantic method is a widely-used technique for text retrieval from a large set of documents [37–39]. Dumais et. al also exhibited that this method is favorable for comparison as it is based on the keyword existing in the documents and also as it does not need human intervention [37]. Another study by Furnas et. al. shows that the latent semantic analysis technique is useful because it scales down the problems associated with the vocabulary mismatch [38].

CHAPTER 3

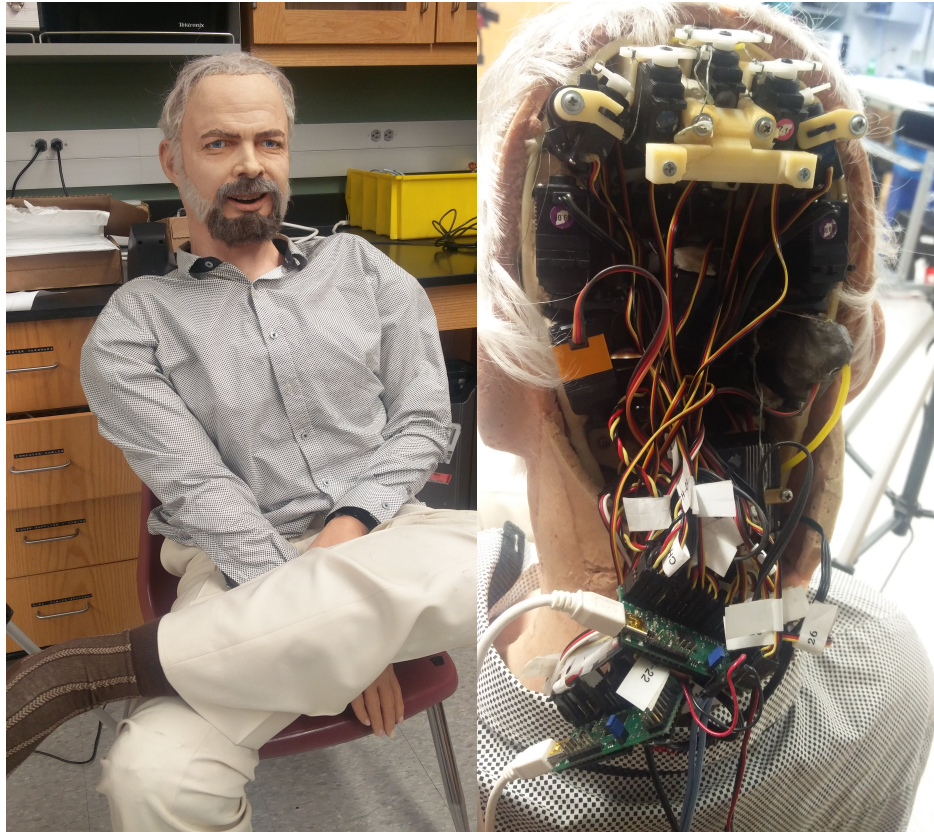
SYSTEM DESCRIPTION

An expressive android robot, named Philip K. Dick or PKD, was used to implement the speech engine developed, in this work while the Baxter robot was used to perform specific tasks as instructed by a user through a tablet interface. An android tablet was used as an interface and to send and receive various requests by a user. Before the software packages developed for the ARNA sitter robot were deployed, the codebase was tested on different robot platforms such as PKD and Baxter. In this section, these platforms as well as the auxiliary devices used for the research work are described in detail.

3.1 PKD Hardware Description

PKD is an expressive android robot which is modeled after the famous science fiction writer Philip K. Dick [40]. Its ability to express various facial features makes it a perfect fit for verbal interaction with human users as well as to express emotion during any conversation. The skin of the robot is made up of Frubber[®], which is a compliant silicon elastomer. Frubber[®] can be easily molded to impart a realistic human-like appearance to the robot face as seen in Figure 5. The facial skin is also able to retain the facial contours and ridges which accentuate the realistic nature of the robot face.

Due to the physical properties and characteristics of Frubber[®], it is widely used to mold robot faces. Some other examples of such robots using Frubber[®] are Hubo [41] and Face [42]. One of the physical properties of Frubber[®] is that it requires less power to be manipulated, pulled or pushed by actuators. The power required is 1/20th of what would be needed to manipulate a standard silicone rubber [43]. This low power requirement makes it ideal for manipulation with servos connected to different points of the skin. The elastic nature of the material prevents the skin from being torn apart during manipulation.



(a)

(b)

Figure 5: PKD in the NGS lab

PKD's head is made up of a thick plastic skull which closely resembles the skeleton of a human head. The plastic skull acts as a housing for the servos which help in manipulating the skin to make facial expressions. To make various facial expressions, 24 points are manipulated on the face skin. The points are as follows: jaw, Procerus, Orbicularis Oculi and Frontalis muscle, cheek region, Duchenne smile, lip region, nasal muscle and eye region. These points represent the muscles of the human anatomy. Along with the actuators for facial expressions, there are 4 motors at the neck to manipulate the heads yaw, pitch and roll movements. All of the motors are connected to 2 Pololu Motor controllers which are connected to the computer. Motor position commands are sent from the computer to the Pololu controller and through the servo movements, the face skin is pulled or pushed. The servos are connected to the Frubber[®] through anchors at the specific points and the

rotational movements of the servo induces pull or push effects. There is also a camera in the left eye of PKD to provide a view from the robot's perspective. This camera is also used for tracking objects and users.

These expressive behaviors of PKD can be used for implementing social interaction modules to enable it to interact with users. During any conversation with a user, it can track the user to maintain eye contact and the facial expressions can convey the mood of the conversation.

3.2 Baxter Robot

The Baxter robot is a new type of collaborative robot with two 7-DOF arms [44]. The maximum reachability of each arm is 1.21 m. This provides the ability to pick and place an object within a slightly longer reach than a human. It weighs about 306 lbs and can lift a payload of 5 lbs. It has cameras embedded in each arm near the wrist area to assist with close range object identification for grasping. It also has embedded force sensors in the arm joints which provide force and torque data as feedback to the controller. Figure 6 shows the Baxter robot used for the purpose.

The Baxter robot has an Ethernet port through which it is connected to the same network as the tablet. This allows the Baxter robot to communicate with the tablet as well as other computing devices through ROS. Baxter runs the Robot Operating System (ROS) and can communicate with external applications through ROS topics. ROS facilitates this distributed computing by making one device serve as the master and other connected devices serve as slaves. Therefore, any data published on a ROS topic can be accessed throughout all the computing devices listening to the ROS topic.

The paddle fingers of the Baxter were used for having more surface area for the gripper. A sheet of silicone layer was attached on the top of the finger for a strong grip of the objects with different shapes as seen in Figure 7. The camera on Baxter's arm can be used for tracking and identifying objects. The camera can be opened and used in several resolution options available but the tracking worked best in 1280x800 resolution [45].



Figure 6: Baxter Robot



Figure 7: Baxter Gripper

3.3 ROS

Robot Operating System (ROS) is a Linux based software platform, which helps in the development of the framework to build robot software. The programming languages used in ROS are C++ and Python [46]. The real-world simulation can be emulated using an application known as Gazebo [47]. The sitter robot uses ROS to create reusable packages required in the project.

ROS caters to the tools and libraries used widely in the robotics world such as object identification, planning, and SLAM (simultaneous localization and mapping), etc. This reduces the efforts of recreating and developing already built packages. Our aim of using ROS is to use these packages for the sitter robot as well as develop new ones to be released.

During this research, several ROS libraries were created for the Baxter robot, including the Cortex module and ROS services for communication between the Tablet interface and the robot. These modules are described in detail in Chapter 4.

URDF (Unified Robot Description Format) was used in building a standardized structure for the robot which is supported by ROS. A URDF can be described as an XML format document defining the robot model such as the kinematics, sensors, and dynamics. This information is useful in the computation of the transforms (tfs) in real-time but it is the pose of the robot in itself and not in the real world. In our project, URDF tress were used for solving the kinematics of the robot.

3.4 Microsoft Kinect[®]

The Microsoft Kinect[®] is a depth camera as shown in Figure 8. It captures an RGB color-stream as well as depth-stream. The components of a Kinect are as follows:

- Vision camera,
- 3D depth sensor
- LED,

- Motorized tilt and
- Microphone arrays.

The vertical and horizontal peripheral vision of the Kinect camera are 43 degrees and 57 degrees, respectively. The rate at which the depth data is streamed is 30 frames per second (FPS). The accuracy of the accelerometer is approximately 1 degree [7].

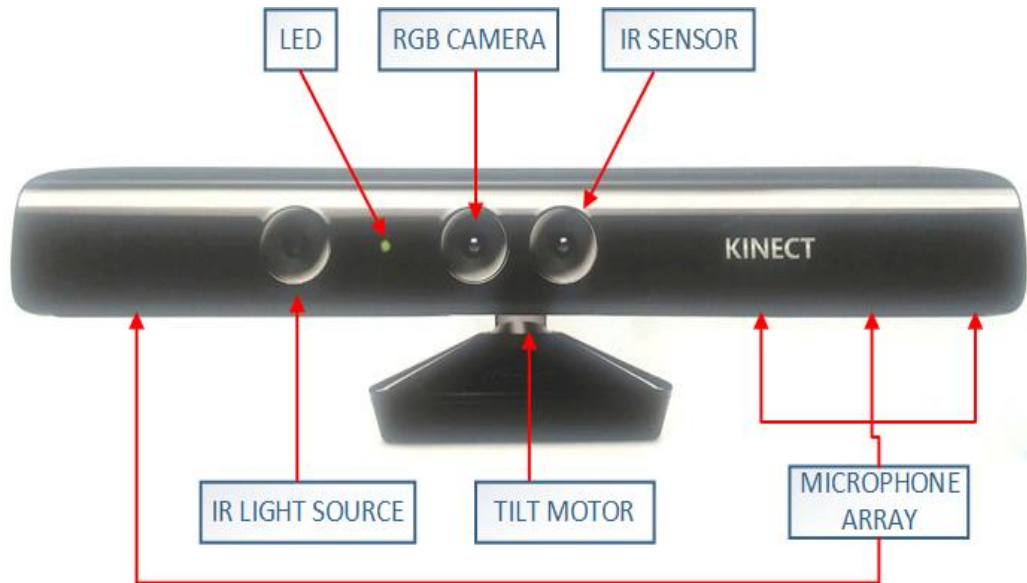


Figure 8: Microsoft Kinect Components [7,8]

The software development kit (SDK) provided by Microsoft helps in skeletal tracking, facial recognition, and seated-mode or far-mode based depth perception. The Kinect was also placed on the head of the Baxter robot as seen in the Figure 9 for capturing the field of view of the robot. This feed is published on to a topic and is subscribed by the tablet node and is shown in the user interface. It is explained in detail in section 3.5.



Figure 9: Kinect Camera on Baxter Head

3.5 Google Pixel C Tablet

The Google Pixel C tablet, as seen in Figure 10, was used to deploy the user interface software package. The graphical user interface is an android app developed in Android Studio 2.3. To interact with ROS, several rosjava libraries were used in the interface for communication. Rosjava is robot independent and provides several libraries to build the front-end user interface and connect it to a ROS network. Once the tablet is connected to the ROS master, it can publish data to ROS topics to command the sitter robot to perform certain tasks. Once some data is published to a ROS topic, the server node running on the master device processes the data and decides on the action to be taken by the robot.

The internal microphone of the tablet is used for the speech interface. The user can speak into the microphone and the speech is then converted to text by running a speech to text algorithm. Once the text is extracted, it is sent to a server for further processing.



Figure 10: Google Pixel C Tablet

The user interface also provides a live video feed from the robot's perspective to help the user ascertain the surroundings of the robot. The user interface also features an abort button, as seen in Figure 11, for stopping the operations immediately. This provides added safety along with the hard E-Stop button.

All of the above-mentioned hardware was to deploy and test the functionality of the software packages intended to be used with the ARNA sitter robot. As seen in Figure 11 the left most part of the GUI contains buttons which are used to give commands. On click of either "PICK AND PLACE" or "PICK AND HANDOVER" button a drop-down menu appears which contains the list of the items that robot can fetch for the user. The center part of the GUI has the mic button to send verbal commands or to have a conversation. On the right bottom of the GUI, options for selecting the grasping orientation of the gripper can be chosen.

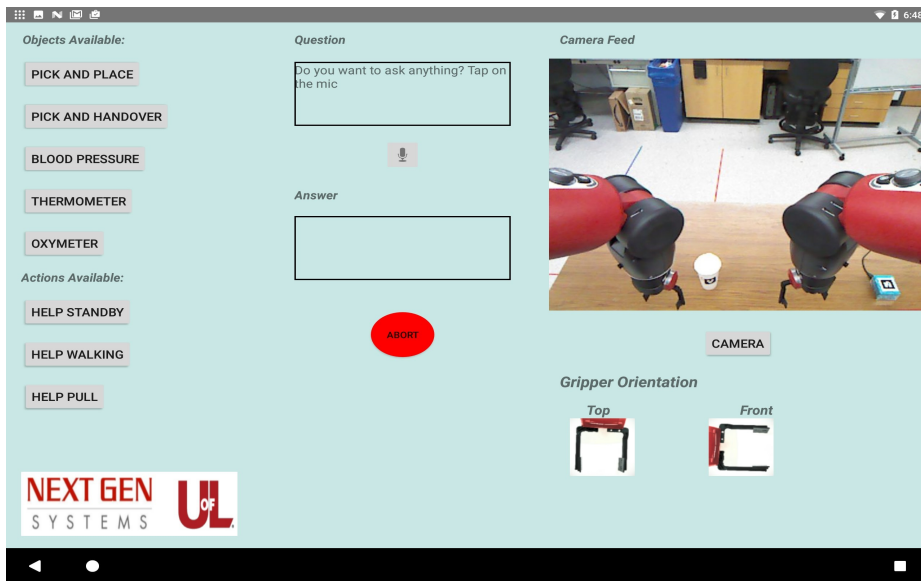


Figure 11: User Interface on Google Pixel C Tablet

The speech engine was also integrated with the PKD robot using the Android tablet app as shown in Figure 12. The GUI contains the speech interface to interact with PKD.

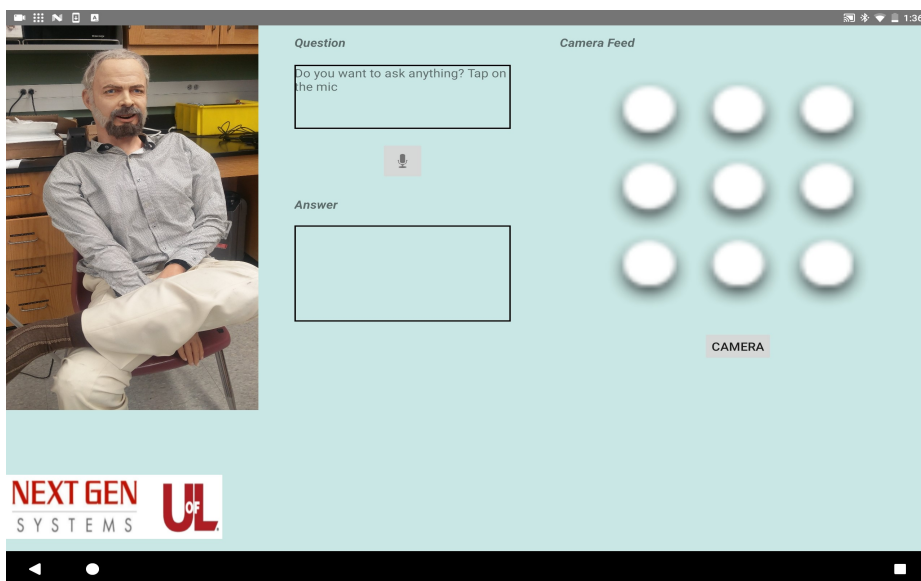


Figure 12: User Interface for interacting with PKD

CHAPTER 4

HUMAN MACHINE INTERFACE FRAMEWORK

An Adaptive Robot Nursing Assistant Sitter Robot or ARNA Sitter robot should assist nurses as well as patients in a patient sitter scenario as identified by Cremer et. al. [9]. In a patient sitter scenario, the mundane and routine tasks which are non-invasive in nature can be performed by a robot. This would be more economic as well as a reliable way to perform the duties. Figure 13 shows the potential hospital room and the patient bed where the patient sitter would be deployed.

The patient monitoring task, as the name suggests, requires the robot to monitor the patient constantly and have the ability to call a nurse in case of any emergency. The task also entails the robot to take measurements of the patient's blood pressure, heart rate, oxygen saturation, body temperature, etc. These measurements are to be taken without any invasive procedure. To take measurements the robot has to estimate the location of the patient and move closer to the patient while picking up the instrument to measure the vitals. Once the readings have been taken, these data can be recorded in the hospital database system. Whenever the nurse or the doctor require to see the reports, the stored data can be pulled up through a user interface. If the readings are out of the normal range, the robot would then immediately alert the nurses and doctors regarding the issue. This would allow them to treat the symptoms as soon as they are detected.

The ARNA Sitter Robot also has to monitor the patient while the patient is resting on the bed. If the patient is about to fall off the bed or trying to get out of the bed when the patient isn't supposed to, the sitter robot should alert the nurses immediately. This would potentially prevent any harm to the patient. Since the patients sometimes get down the bed on their own against medical instructions, the sitter robot would be able to detect such occurrences. While alerting the caregivers, the sitter robot would also issue vocal warning

to the patient about the potential danger of getting out of the bed.



Figure 13: Hospital room and patient bed [9]

During the sitting sessions, the patient may also ask the sitter robot to fetch items from the room. The sitter robot has to search for the item, pick up the item and hand it over to the patient to complete the pick and place task. To identify items, the robot can make use of the QR-code attached to the items. To decide the grasp strategy, such as grasping from top or front, the user may instruct the robot the correct grasping pose. Using this information from the patient or user, the robot can scan the workspace, pick, and place the required item.

All of these interactions can be achieved through a handheld device such as a tablet. A tablet interface can provide both an onscreen button based interface as well as speech interface. The patient can control the sitter robot through these interfaces depending on the level of ease of use. If the patient is not comfortable using the onscreen buttons or holding the tablet, the speech interface may be used to issue verbal commands to the robot. The tablet also provides access to converse with the robot. The conversational abilities are provided through a chatbot based server where the tablet app sends the conversation requests and the response is sent from the server. This conversational ability can keep a patient engaged.

The chatbot web server takes SOAP requests. These requests are processed through the ALICE chatbot based algorithm. The AIML files make up for the knowledge base and the request is searched against the knowledge based to find the closest response. If the context of the request is not present in the AIML knowledge base, the server then looks for an appropriate response in the Wikipedia based knowledge base which consists of Wikipedia article sentences that have been processed using Latent Semantic Analysis (LSA). After the correct context is found, the response is then sent back to the client. This process helps the robot respond in the context of the conversation, and avoids irrelevant responses from the robot.

To implement the pick and place task and the conversational abilities, we have used the Baxter robot and the Google Pixel C tablet. The tablet was loaded with the user interface to be used in the sitter robot scenario. The Baxter robot was interfaced with the system architecture to perform the tasks of pick and place or pick and handover requests initiated from the tablet. The requests may be verbal or onscreen button based in nature.

The research work presented here focuses on providing a reliable interface framework between a patient or a nurse and the sitter robot. The HMI proposed is implemented as an Android application on a tablet. This interface accepts both voices as well as on screen button press commands and relay high-level user commands to the robot. The sitter robot can perform certain routine tasks such as pick-and-place objects at certain locations and pick-and-handover objects to the user. The interface is also capable of socially conversing with the patient. Since this HMI framework is intended to be used in a healthcare facility environment and the user base includes the patients and nurses, the user is allowed to issue emergency stop commands during robot operation. Such commands are given priority and allow the robot to return to a safe position. In certain situations, the patients may not be able to hold or operate the tablet using onscreen buttons. They can still interact with the robot by using the speech interface provided by the tablet application. Figure 14 shows an overview of the different systems and their communication with each other in order to complete the framework.

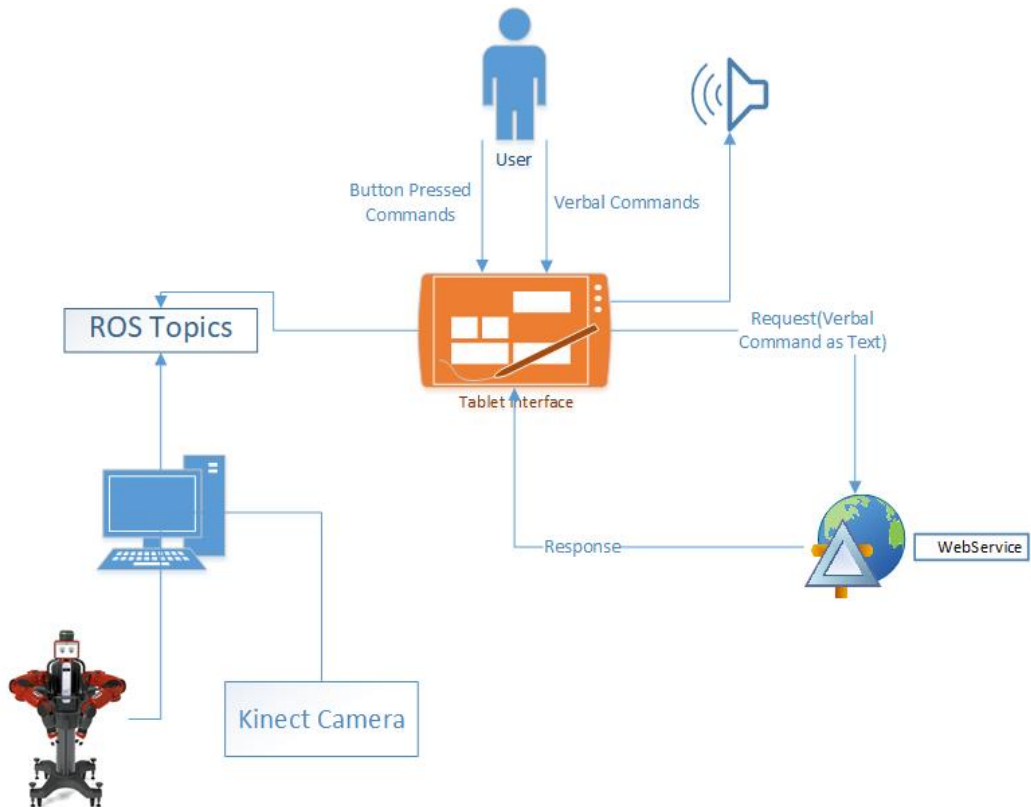


Figure 14: System Overview

4.1 System Architecture

The Human Machine Interface software was developed by logically separating the low-level robot specific control modules from the high-level perception, planning, and manipulation modules (also called the Cortex modules). This makes the framework compatible with many different robots and input devices. It also allows the high-level Cortex algorithms to be reused without knowledge of low-level controllers. Figure 15 depicts the interaction between the HMI and the Robot Cortex, which is ultimately responsible for traded control and autonomous skills for the robot. Currently, the Cortex can assist users by picking-and-placing objects, having conversations and listening to pick-and-place commands to trigger appropriate routines:

- The pick-and-handover module can pick an intended item and handover the item to the user standing in the vicinity of the robot.

- The speech interface of the framework listens to voice inputs from the user and calculates whether the speech input was meant for conversational purposes or to issue pick-and-place commands.
- The framework also has provisions to issue a software emergency stop to disable robot operation.

As the intended users of this interface are nurses or patients in a healthcare facility, the speech interface and onscreen buttons provide options to a wide user base who may not be able to hold the tablet for a long period of time. A state machine constantly checks if there are any stop flags set by the tablet interface, in which case, it ceases the robot movement. It also checks for the continuous connectivity between the tablet and the robot.

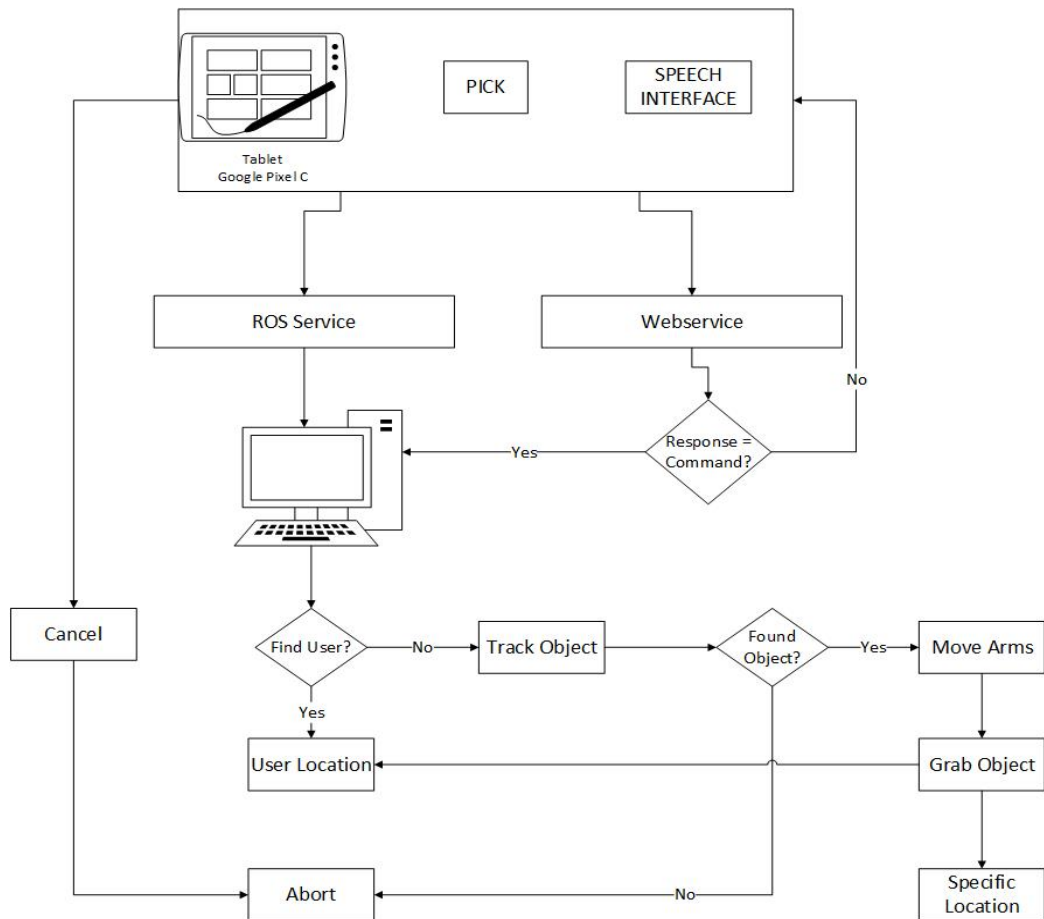


Figure 15: Proposed Human Machine Interface Framework

The proposed framework implements a traded control scheme which recuses the user from actively driving the robot throughout the item retrieval process. This is an important feature in environments where teleoperation of a robot is not recommended for untrained users. The traded control scheme takes inputs from the users in the form of verbal or gesture commands and autonomously performs the tasks based on the environment.

In pick-and-place scenarios, the user identifies an object from the list of items suggested by the tablet application. The user can either push the onscreen button for the item or speak into the tablet. Upon receiving the command from the user, the Cortex identifies the desired object through an RGBD camera sensor. If the object is not found, an error message is displayed on the screen of the tablet stating the issue. Once the desired object is found, the arm trajectory planner plans the motion of the arm to reach the coordinates calculated by the object detection module. If the arm trajectory planning succeeds, the joint angles are then provided to the arms and the arm tip is actuated to the location of the object. If the planning fails, as it is sometimes the case with many off-the-shelf ROS planners, the process is restarted until/before it will time out. The desired object is then grasped and placed at a drop-zone. The pick-and-place module can be combined with the user identification module to create the pick-and-handover module. In pick-and-handover item module, the robot searches for the user and hands over the item by extending its arms.

Safety features integrated into the framework modules are activated when the user is not found, the intended object to pick is not found, or the user issues a task cancel command. During the pick and handover routine, if the user is not detected or not present, an error flag is set. The error flag initiates an error handling routine which retraces the trajectory that the robot arm followed after picking up the object. It allows the robot to return the object to its original position. After the object is returned safely, the robot resets to its rest position. When a task is initiated by the user but is then cancelled mid-routine, the task cancelled routine is initiated. In this routine, the progress of the state machine is halted and any movement performed is rolled back. The robot then returns to its resting position. These conditions can also be visualized in Figure 15.

For manipulating Baxter’s arm, the Moveit! motion planning framework was used. Moveit accepts a Cartesian location and plans the arm trajectory. Once the trajectory planner succeeds, the planned motion is executed through joint angle manipulation. This is used during the pick and place activity, upon the request from the tablet. The details of the object to pick are sent as a request from the tablet. The service request triggers an object identification routine. Once the object is identified, its location with respect to the wrist of Baxter is calculated. This provides the Cartesian difference between the wrist and Baxter. These calculated values are then used by Moveit for trajectory planning of the arm.

The object identification process was carried out through the QR code detection method. For detection, we use Baxter’s on-board wrist cameras. On each object in Baxter’s workspace, a QR code is attached. In order to find an object, Baxter’s arm first scans the workspace to detect the required QR code associated with the object of interest. Next, the location of the QR codes with respect to Baxter’s frame of reference is determined and are input to the Moveit module to plan the arm trajectory.

The Kinect installed on the head of Baxter is also used for user detection. This is achieved by detecting the QR code attached to the tablet held by the user. The head movement of Baxter enables the framework to scan the surroundings for the user by rotating the head. This enables the pick and handover module to calculate an estimated coordinate to reach after the object picking is completed.

4.2 Integrating Tablet and ROS

The Android application was developed using Android Studio. Along with Android Studio, the Android software development kit and rosjava were used to interface ROS with the tablet application. Rosjava contains libraries which are required to establish communication and exchange data between the tablet and a ROS system. Android Studio 2.3 was used to develop the tablet application. The android application is separated into 3 different classes: MainActivity, NodePublisher, CallWebservice and the AbortPublisher.

- MainActivity: The MainActivity is the main class which links the graphical user

interface (GUI) building blocks such as buttons, text view, etc. to their functionality. It also has the operation of converting speech to text.

- **NodePublisher:** The NodePublisher connects the GUI to the Robotic Operating System(ROS) running on the robot using ROS service. The commands published to the ROS service by this class is any operation performed by the user on the GUI or by using the speech interface.
- **CallWebservice** connects the GUI to the Simple Object Access Protocol (SOAP) web service. It sends the text received from the speech to text interface to the web service through SOAP request. The web service differentiates the text as command or conversation. If the response from the web service is robot specific commands, then it is redirected to NodePublisher to publish the command else it is transformed using the text to speech interface to have a conversation.
- **AbortPublisher** publishes the message to the ROS topic if the “Abort” button the GUI is pressed. If there is an ongoing task, then that task will be stopped and the abort state will be achieved. This message is prioritized to have safer pHRI.

The SOAP Web service was implemented in Java using the Eclipse IDE and deployed on localhost using Tomcat Apache server 9.0. The workstation in which the web service was deployed locally was connected to the same Wi-Fi network as the tablet and the Baxter robot. It is based on the concept of a natural language processing chat engine Artificial Linguistic Internet Computer Entity (ALICE), also referred as Alicebot. The knowledge base of the chat engine is composed of question answer modules, named categories and described by Artificial Intelligence Markup Language (AIML). User questions are compared with the patterns stored in the knowledge base.

If the response is not found in the AIML pattern, then the other knowledge base created using Wikipedia resources is used to find a suitable response. Apache Solr which is an open source full-text search engine on the database. The database was built using Wikipedia xml resources. Once the response is found to the user input, it is transmitted

back as a SOAP response to the client request. This response is processed in the Android application and displayed in the response text box. It is then converted from Text to Speech using the Android libraries.

CHAPTER 5

SPEECH INTERFACE

This chapter covers the speech interface of the nursing robot. The interface was deployed as a web service and developed using the Java and Eclipse Integrated Development Environment (IDE). It consists of two knowledge bases. The first knowledge base comprises the basic Artificial intelligence markup language (AIML) tags and the second one consists of sentences from Wikipedia articles.

5.1 Web service

A Web Service can be described as a service that can be accessed from any system over the web. There are two types of web services: REST (Representational State Transfer) and SOAP (Simple Object Access Protocol). We created a SOAP web service which uses the XML format based messaging protocol. The web service was created using the Java programming language in Eclipse and deployed on localhost using Apache Tomcat Server version 9.0.

The skeleton of a SOAP message, which is XML based, can be visualized in Figure 16 and each component of the message is defined as follows:

- Envelope: The SOAP envelope helps in determining that an XML document is a SOAP message.
- Header: The header is contained in an envelope and is an optional entity. The content of the header imparts information regarding authentication, concealing/encoding the data, or how to decode the message received by the client.
- Body: Another part of the envelope is the body that holds the information.

- Fault: This is also an optional entity and it gives information about the errors prevailing during the processing of the request.



Figure 16: SOAP message skeleton

The intercommunication between the tablet and the web service was performed through the medium of Hypertext Transfer Protocol (HTTP), as illustrated in Figure 17. HTTP is the basic code of conduct for communication used by the World Wide Web. It describes the mechanism of how to format and transport the message, and the procedure web servers adhere to while responding to numerous commands.

When the patient clicks on the mic button in the tablet interface and speaks, that speech is converted to text using the internal Android SDK in Android Studio. That text is then wrapped in a SOAP envelope to generate a SOAP request. The SOAP request is transported to the web service using HTTP transport. Once the web service completes the processing of the SOAP request, a SOAP response is generated. This response is sent back to the tablet interface using the HTTP transport. Then further parsing of the response is done to get the relevant information. This information is then converted from text to

speech using the internal Android SDK libraries.

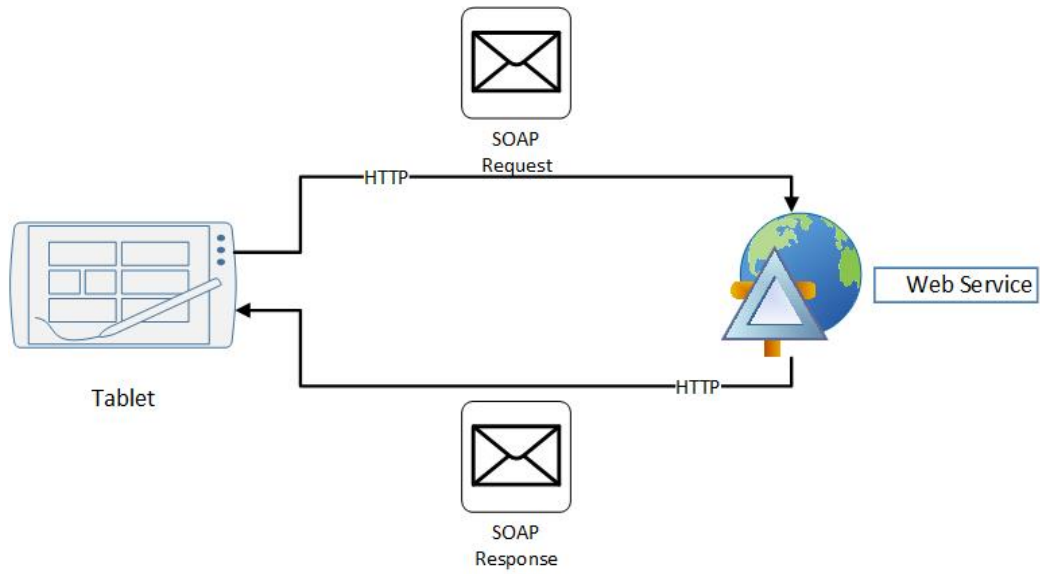


Figure 17: SOAP Web service HTTP Protocol

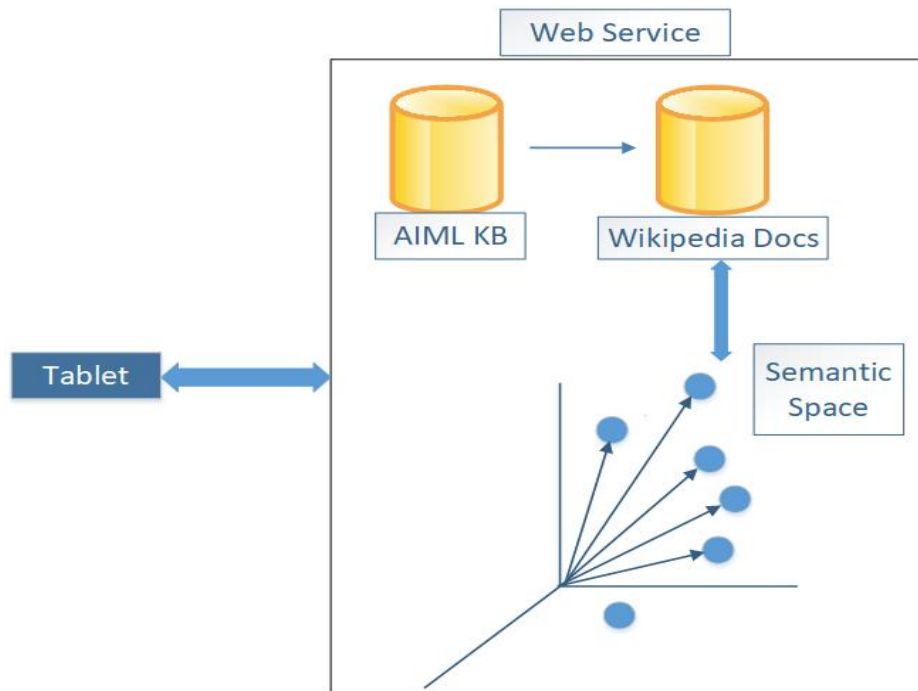


Figure 18: Web service Flowchart

When a request is sent from the tablet, the web service first looks for a response in the AIML knowledge base. If the AIML knowledge base does not have an answer to

the query asked by the user, it searches the Wikipedia document’s semantic space for the document that is most relevant to the query and this document is given as an answer. This overview of the web service flow is illustrated in Figure 18.

5.2 AIML Knowledge Base

The Artificial Linguistic Internet Computer Entity (ALICE), alias Alicebot, was the first chatbot created whose brain was based on AIML. Alicebot is a free software available over the internet and its knowledge base can be modified as per the requirement. AIML is a constituent of several entities. Some of the entities used are described in brief in table 1.

TABLE 1: AIML tags description

AIML Tag	Description
<aiml>	Defines the start and the end of the AIML document.
<topic name = “XYZ”>	Helps in keeping track of the subject of the conversation which would help in the chats coming after.
<category>	The basic unit of the knowledge base.
<pattern>	Compares and matches the input sequence to the pattern.
<that>	Keeps track of the previous response depending on the context.
<template>	The template of the AIML response to the input pattern.
<sr>	This is a shortcut tag for the <srail>
<sentence>	Transforms the first word to uppercase.

<code><condition></code>	Helps in breaking down the input sequence that matches for better response. This is analogous to a switch-case in a programming language.
<code><random></code>	Helps in giving different or varied responses to the same pattern.
<code><srai></code>	This is a recursive tag used for comparing other categories.
<code></code>	Identifies a list item used inside the random tag and is used in conjunction with random and conditions to define different responses.
<code><think></code>	It hides the value of a variable it saves from the user.

Among all the AIML tags described above, we will be discussing the four important tags in detail and how they are implemented.

- `<aiml>` this tag specifies the start and the finish of an AIML document. The tag can also have information regarding the version and encoding but it is optional. Each `<aiml>` tag must have one `<category>` tag. We can have numerous AIML documents in the knowledge base but each document must have only one `<aiml>` tag. For example, one AIML document should look as follows:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<aiml version = "1.0.0" encoding = "UTF-8"?>
<category>
. . .
```

```
</category>
```

```
</aiml>
```

- `<category>` tag is the basic knowledge entity of the AIML document. Each such tag should contain `<pattern>` and `<template>` tags. So, it should look as follows:

```
<category>
```

```
<pattern>. . . </pattern>
```

```
<template>. . . </template>
```

```
</category>
```

- `<pattern>` tag contains the input to the chatbot interface from the user. In order to match multiple user inputs to the same pattern tag, a wild card is used. Inside `<category>` tag, this tag has to be the first tag. AIML is not case sensitive. So, if the user input is “hello” or “HELLO”, it would be matched to the same pattern containing “HELLO”. The text in the `<pattern>` tag should always be in caps.

```
<category>
```

```
<pattern>HELLO BAXTER </pattern>
```

```
<template>. . . </template>
```

```
</category>
```

- `<template>` tag stores the response to the user query. Inside `<category>` tag, this has to be the second tag. Once the `<pattern>` has been matched with the user’s input, the data stored in the `<template>` tag is used as the response. The `<template>` tag can also initiate another program or pass on to other categories using the `<srai>` tag.

```
<category>
```

```
<pattern>HELLO BAXTER </pattern>
```

```
<template>Hi There. How are you? </template>
```

```
</category>
```

An example of the AIML knowledge base implemented can be seen in Figure 19. The figure shows the pattern “what is socialism”. So, if the user input is the same as

this pattern, the template response is chosen. Note that the template tag has another tag called `<random>` tag which means that the response can be any of a random list inside the `` tags.

Another example of AIML, using the `<srai>` tag, is shown in Figure 20. When the input matches the pattern “DO YOU HAVE HUMOR”, the `srai` tag inside the template changes the pattern to “DO YOU HAVE A SENSE OF HUMOR” and checks for other categories which have the modified pattern.

```
<category>
  <pattern>WHAT IS SOCIALISM</pattern>
  <template>
    <random>
      <li>Communism from people who want to keep their Volvos.</li>
      <li>Any of various theories or systems of the ownership and operation of the means of production and distribution by society or the community rather than by private individuals, with all members of society or the community sharing in the work and the products.</li>
    </random>
  </template>
</category>
```

Figure 19: Example of AIML

```
<category>
  <pattern>DO YOU HAVE HUMOR</pattern>
  <template>
    <srai>DO YOU HAVE A SENSE OF HUMOR</srai>
  </template>
</category>
```

Figure 20: Example of `<srai>` tag in AIML

5.3 Wikipedia Knowledge Base

Wikipedia consists of articles from a wide range of topics and it is the largest encyclopedia available online. The entire set of Wikipedia articles or wikis are available in the

form of XML documents. These documents were downloaded from the Wikimedia site, and used to create a second knowledge base for the speech engine.

WikiExtractor was used to generate plain text from the Wikipedia XML dumps. These plain text files were used to generate the semantic space for the knowledge base. Later a few documents were added from the online forums. The semantic space was created using the latent semantic analysis (LSA) technique.

5.3.1 Latent Semantic Analysis(LSA)

LSA is a mathematical approach of defining a relationship between a set of documents and the words contained in those documents.

Suppose we take n documents and create a document-word matrix X such that the element $x_{(i,j)}$ refers to the frequency or the number of matches of term i in the document j .

$$\begin{array}{c}
 d_j \\
 \downarrow \\
 (t_i)^T \rightarrow \begin{bmatrix} x_{1,1} & \cdots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \cdots & x_{m,n} \end{bmatrix}
 \end{array} \tag{1}$$

Hence each row in the matrix represents the vector corresponding to the term and its association to documents. Similarly, each column of the matrix is a vector that corresponds to a document and its relation to each term.

The correlation between the terms over a set of documents is given by the dot product of the term vectors. And the dot product between two document vectors gives a correlation between documents over the terms and is given by:

$$(d_j)^T d_q = (d_q)^T d_j \tag{2}$$

In linear algebra, the singular value decomposition of X can be done by factoring or decomposing X into a matrix multiplication of matrices U and V which are orthogonal

matrices and Σ which is a diagonal matrix. Hence, singular value decomposition (SVD) of X is given by:

$$X = U\Sigma(V)^T \quad (3)$$

Dimensionality reduction is a technique to convert the terms that exist in a high-dimensional space into a lower-dimensional space. If we take k as the dimensionality reduction value, then we get the corresponding singular vectors U and V , and also we get the rank k approximation to X . The approximation can be represented as

$$X_k = U_k(\Sigma)_k(V_k)^T \quad (4)$$

After the formation of the vector representation of the documents in semantic space, the similarity between a query q and document d_j is calculated using the cosine similarity, as shown below:

$$sim(d_j, q) = \frac{d_j \cdot q}{\|d_j\| \|q\|} \quad (5)$$

The “ \cdot ” is the dot product between the vectors and “ $\|\cdot\|$ ” is the 2-norm representation of the vectors.

The Text Mining Library (TML) for LSA [48], which is a Java library, was used to implement the semantic space.

CHAPTER 6

EXPERIMENTS AND RESULTS

6.1 Human Machine Interface

Preliminary experiments were carried out to test the functionality of the framework and the tablet app. Tests were conducted with non-expert users, who had no prior familiarity with the system, as well as expert users (programmers on similar systems in our lab). First, users were briefed about the limitations and the scope of the tasks that can be performed through the tablet interface, and the scenario that the patient sitter will be performing. After the briefing, the users interacted with the robot while a supervisor was present to stop the experiment should any issues occur. A cup, box, pill box and a bottle were placed randomly on a table in front of the robot. All the items have markers attached to them, which are visible to the robot in the poses of interest as seen in Figure 21. These markers assist the robot in identifying the location of the items during manipulation.



Figure 21: Markers for object identification

In the first set of experiments, the users issued commands by clicking on the on-screen buttons and the success rate for completion of the tasks were recorded. The next set of experiments involved users issuing verbal commands for the same pick and place tasks. The verbal commands were not specific but conveyed the same message. For a robot to recognize the message as a robot command, the message should contain the object name and identifiers such as “get”, “fetch”, “bring”, and “need”, etc. Some of the examples of the verbal commands issued to fetch a cup are Get me a cup, fetch me a cup, can you get me a cup, I need a cup, bring me a cup etc. The success rate of identification of the command and completion of the tasks were recorded. Finally, we tested the safety of the interface by asking the users to initiate task abort commands through voice as well as on-screen buttons. The success rate of the emergency stop process was recorded. The following metrics were considered for measuring the performance of the tablet interface for each experiment:

- Ease of Use, as measured by a questionnaire.
- Time to complete pick and place task.
- Conversation through speech interface rated by users.
- Success rate in pick and place task.
- Command abort success rate.

The experiments were conducted by using two untrained users and two expert users. Twelve trials for each user were conducted, six verbal and six on screen button commands were issued. During experimentation, the users directed the robot to grab any item, from the cup, bottle and the box, in random order as they please and the robot picked up the items and placed it in a designated drop zone. The commands were first sent through the on-screen buttons and the whole process was repeated with users issuing voice commands by framing any sentence as per their discretion to convey the sense that they want a certain item.

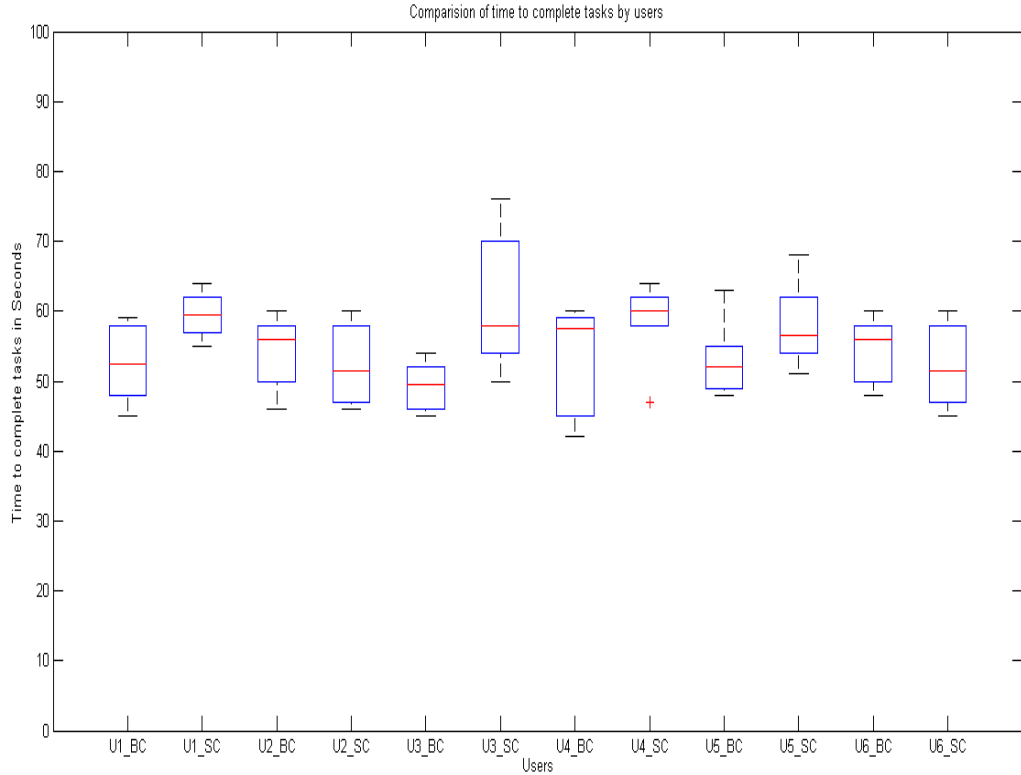


Figure 22: Time of completion comparison between users

During this process, the total times from the issuing of commands to completion of tasks were recorded along with the time taken by the robot to recover from failures (grasp, motion planning, etc.). The users were then directed to talk to the interface and rate their experiences accordingly. Results are shown in Figure 22, where U1 and U2 represent the expert or trained users and U3, U4, U5 and U6 represent the untrained users. The label BC refers to button commands issued by a user and the label SC represents the speech commands initiated by a user. It was observed that for trained users, the average completion time of the tasks was 54.6 ± 5.05 seconds. For untrained users, after an initial training period of 90 seconds, the operation time for subsequent trials went down to an average time of 55.04 ± 6.24 seconds which is comparable with the trained users. This underlines the fast training and usability of our HMI framework.

Users were also provided with the following questionnaire to rate their experience with the application. Each question had options to answer based on a Likert scale with

TABLE 2
Average rating of the interface by the users

Users	Ease of Interface	Speech Interface
Trained User 1	4/5	4.5/5
Trained User 2	4.5/5	3.8/5
Untrained User 1	4.5/5	4.0/5
Untrained User 2	4.6/5	4.2/5
Untrained User 3	4.5/5	4.1/5
Untrained User 4	4.7/5	4.3/5

five-level Likert items.

- Did the speech interface recognize what you said correctly?
- Was the response valid to a question asked by you?
- Did you have to raise your voice pitch in order to interact with the speech interface?
- Was the audio response from the interface clear?
- Was the application interface easy to use?
- Was the speech interface helpful to you?

During experiments, it was observed that in certain poses inverse kinematics calculations of Moveit for the Baxter arm did not succeed. When such conditions occur, the interface directs the robot to move to a known stable location and retry planning the trajectory to complete the task. For the speech interface, it was observed that accurate detection of the spoken words varies depending on the accent of the person talking to the interface. This leads to a lower speech usability rating than that of the tablet interface as noted in table 2.

6.2 Speech Engine

The LSA produced knowledge base was tested by probing the system with different speech-based test queries. The dimensionality reduction for LSA varied from 50 to 200 in

increments of 50 (hence 50, 100, 150, and 200), and each query was submitted against the semantic space obtained using these four different dimensions to evaluate the retrieval using these different reduced dimensional semantic spaces. The first three responses were checked to see if the response exists for the query.

The test queries were divided into easy, medium and hard based on the complexity of the question. Below, we list for each category (easy, medium, hard), the query (question) and the responses (answer) obtained for each of the four tested reduced dimensions of LSA.

TABLE 3: Easy Questions

Question No.	Question	Dimensions Reduction	Answer
1	What is Chemotherapy	50	Chemotherapy is a category of cancer treatment that uses one or more anti-cancer drugs as part of a standardized chemotherapy regimen.
		100	Chemotherapy is a category of cancer treatment that uses one or more anti-cancer drugs as part of a standardized chemotherapy regimen.
		150	Chemotherapy is a category of cancer treatment that uses one or more anti-cancer drugs as part of a standardized chemotherapy regimen.
		200	Chemotherapy is a category of cancer treatment that uses one or more anti-cancer drugs as part of a standardized chemotherapy regimen.

2	What is BMI	50	Body mass index (BMI) is a measure of body fat based on your weight in relation to your height, and applies to most adult men and women aged 20 and over.
		100	Body mass index (BMI) is a measure of body fat based on your weight in relation to your height, and applies to most adult men and women aged 20 and over.
		150	Body mass index (BMI) is a measure of body fat based on your weight in relation to your height, and applies to most adult men and women aged 20 and over.
		200	Body mass index (BMI) is a measure of body fat based on your weight in relation to your height, and applies to most adult men and women aged 20 and over.
3	What are the side effects of Advil	50	Common Advil side effects may include upset stomach, mild heartburn, nausea, vomiting, bloating, gas, diarrhea, constipation, dizziness, headache, nervousness, mild itching or rash or ringing in your ears.
		100	Common Advil side effects may include upset stomach, mild heartburn, nausea, vomiting, bloating, gas, diarrhea, constipation, dizziness, headache, nervousness, mild itching or rash or ringing in your ears.

		150	Common Advil side effects may include upset stomach, mild heartburn, nausea, vomiting, bloating, gas, diarrhea, constipation, dizziness, headache, nervousness, mild itching or rash or ringing in your ears.
		200	Common Advil side effects may include upset stomach, mild heartburn, nausea, vomiting, bloating, gas, diarrhea, constipation, dizziness, headache, nervousness, mild itching or rash or ringing in your ears.

TABLE 4: Medium Questions

Question No.	Question	Dimensions Reduction	Answer
1	What is a lymph gland	50	A lymph node or lymph gland, is an ovoid or kidney-shaped organ of the lymphatic system, and of the adaptive immune system, that is widely present throughout the body.
		100	A lymph node or lymph gland, is an ovoid or kidney-shaped organ of the lymphatic system, and of the adaptive immune system, that is widely present throughout the body.

		150	A lymph node or lymph gland, is an ovoid or kidney-shaped organ of the lymphatic system, and of the adaptive immune system, that is widely present throughout the body.
		200	A lymph node or lymph gland, is an ovoid or kidney-shaped organ of the lymphatic system, and of the adaptive immune system, that is widely present throughout the body.
2	What is normal human blood pressure	50	Normal blood pressure should be more than 90 over 60 and less than 120 over 80.
		100	Normal blood pressure should be more than 90 over 60 and less than 120 over 80.
		150	Normal blood pressure should be more than 90 over 60 and less than 120 over 80.
		200	Normal blood pressure should be more than 90 over 60 and less than 120 over 80.
3	What is normal heart rate.	50	A normal resting heart rate for adults ranges from 60 to 100 beats a minute.
		100	A normal resting heart rate for adults ranges from 60 to 100 beats a minute.
		150	A normal resting heart rate for adults ranges from 60 to 100 beats a minute.

		200	A normal resting heart rate for adults ranges from 60 to 100 beats a minute.
4	What is the normal body temperature.	50	The normal human body temperature range is typically stated as 36.5-37.5 C.
		100	The normal human body temperature range is typically stated as 36.5-37.5 C.
		150	The normal human body temperature range is typically stated as 36.5-37.5 C.
		200	The normal human body temperature range is typically stated as 36.5-37.5 C.

TABLE 5: Hard Questions

Question No.	Question	Dimensions Reduction	Answer
1	What are the symptoms of breast cancer.	50	A breast cancer is cancer that develops from breast tissue. Signs of breast cancer may include a lump in the breast, a change in breast shape, dimpling of the skin, fluid coming from the nipple, or a red scaly patch of skin.
		100	Breast cancer is cancer that develops from breast tissue. Signs of breast cancer may include a lump in the breast, a change in breast shape, dimpling of the skin, fluid coming from the nipple, or a red scaly patch of skin.

		150	Breast cancer is cancer that develops from breast tissue. Signs of breast cancer may include a lump in the breast, a change in breast shape, dimpling of the skin, fluid coming from the nipple, or a red scaly patch of skin.
		200	Breast cancer is cancer that develops from breast tissue. Signs of breast cancer may include a lump in the breast, a change in breast shape, dimpling of the skin, fluid coming from the nipple, or a red scaly patch of skin.
2	What causes lung cancer	50	Pancreatic cancer is the malignant tumor arising mostly from the epithelium of the pancreatic ducts, therefore most of them are adenocarcinomas.
		100	Lung cancer in non-smokers can be caused by exposure to radon, secondhand smoke, air pollution, or other factors.
		150	Lung cancer in non-smokers can be caused by exposure to radon, secondhand smoke, air pollution, or other factors.
		200	Lung cancer in non-smokers can be caused by exposure to radon, secondhand smoke, air pollution, or other factors.

As seen in Table 6, Response 1, Response 2 and Response 3 represent the responses

to the queries sorted in the order of their similarity to the query. Response 1, with dimensionality reduction 50, has a precision value of 70% which means that if we have 10 queries then only 7 queries resulted in the first response or the response with the highest similarity index as the correct response. We notice that as the dimensionality reduction increased from 50 to 100, 150 and finally 200, the probability of getting the correct response from the highest similarity index increases.

TABLE 6: LSA Result

LSA Dimensionality Reduction	50	100	150	200
Response 1	70%	90%	90%	90%
Response 2	80%	100%	100%	100%
Response 3	100%	100%	100%	100%

Analysis: The responses to the queries with dimensionality reduction 50 show that for this specific document collection, projecting onto much lower dimension(50) can result in loss of information as several unrelated concepts (example: lung and pancreatic cancer) get fused into one concept. When this happens, these concepts, occurring in a query and a response, will result in an incorrect match. This in turn would return a wrong or non-relevant response, and would result in low precision or specificity. A higher dimensionality has less risk of fusing different concepts, and thus results in a more faithful projection. However, the downside of a dimensionality that is too high may be that concepts that are indeed related, end up being mapped to different dimensions and thus different concepts. This in turn would preclude matching such concepts when they occur in a query and a relevant response. This situation results in low recall. This illustrates and motivates the need to choose an optimal dimension in LSA to achieve accurate (specific) retrieval. We also note that Question 1, containing the word “symptoms” results in a relevant response, although the response does not contain “symptom” but rather contains a synonym word,

namely “sign”. This match between two distinct, and yet semantically similar words (symptom and sign), in addition to the presence of the word cancer in both query and response, results in a correct response, identified as the most similar to the query.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Conclusion

The work presented in this thesis pertains to development of an assistive robot's (ARNA) social interaction capabilities and control interfaces. The assistive robot's task is to assist nurses in their tasks in a health-care environment. The robot's user base includes both the caregivers as well as the patients. Keeping these objectives in sight, the thesis concentrates on building a reliable system to have a conversation with and developing an intuitive user interface to control the robot.

In order to incorporate the developed systems, a human-machine interface framework was proposed. This framework takes into consideration the user base and facilitates a seamless operation between the user and the robot.

7.1.1 Human-Machine Interface (HMI) Framework

The HMI framework can be logically divided into three aspects, namely the traded control, tablet interface and speech interface. As the user-base for the nursing assistant robot are the health-care facilitators, nurses and the patients, the framework was built, while keeping in mind the desire to make it intuitive enough to reduce any learning necessary to control the robot.

These parameters led to the choice of using a traded control scheme for controlling the robot and instructing it to perform certain routine tasks. The traded control scheme, incorporated in the framework, takes high-level instructions from the user and the low-level planning is done autonomously by the robot. This relieves the user from constantly monitoring the robot during the whole process of task completion and avoids fatigue experienced

by the user during constant teleoperation of the robot.

The framework also includes a software emergency stop button or an E-Stop button which when pressed by the user, disables the robot. This feature ensure that the robot is stopped at any point of operation if something does not go according to the expected plan. This is a necessary feature as the robot works in an environment shared by human users and facilitates the safety of the users.

In the proposed framework, a tablet serves as the human-machine interface. Initial studies were conducted to test the usability of the framework and the level of learning required by a new user before being able to operate the robot. The results from these studies indicate that rate of completion of the assigned tasks by an untrained user and an expert user are quite similar.

7.1.2 Tablet Interface

The tablet interface incorporates all the available services from the proposed HMI framework. It was developed as an Android application that can be deployed on any android devices. The tablet application allows the user to initiate tasks such as picking a specified object and placing it at a specified location, picking specified objects and handing them over to the user, taking vital measurements of the patient such as blood pressure, body temperature and heart rate monitoring.

During pick and place tasks, the user also specifies the method of grasping the object. This allows the robot to pick up unmodeled objects as the grasping strategy is supplied by the user. This reduces the computing time for planning the grasping approach and decision making for an object which is not in the system database.

The tablet application offers two modes of input from the user, i.e. through onscreen buttons and through a speech interface. The tablet application also implements a camera feed directly from the robot which allows the user to view the environment through the robot's perspective. The E-Stop button is also available through this tablet interface which when pressed, halts the robot.

The initial experiments were conducted using a Google[®] Pixel tablet which had the developed application installed on it. The preliminary data from the experiments suggests the ease of use of the developed application by both trained and untrained users.

7.1.3 Speech Interface

The speech interface was developed as a web service and implemented using Eclipse and Apache Tomcat based servers. Deploying the system as a web service allows it to be interfaced with any external system through SOAP requests. Once deployed, the speech interface system was integrated with the tablet interface that was developed. The tablet interface converts the spoken words into text through a speech to text engine and passes the text as a request to the web service. The web service compares the request to its knowledge base and sends an appropriate response.

The knowledge base was implemented using the AIML and Wikipedia knowledge bases. The Wikipedia documents were processed by LSA to create a lower dimensional semantic space that was then used to compare the projected documents to incoming requests/queries, and an appropriate response was chosen based on the highest cosine similarity score between a document in the semantic space and the request.

The experiments conducted during the work concentrate on establishing a relation between the accuracy of the response and the reduction of dimensionality of the knowledge base corpus. It was observed that accurate responses were returned by the web service when the knowledge base corpus was of higher dimension.

Thus, a higher dimensionality of the knowledge corpus was implemented with the proposed HMI framework. This allowed the user to interact with the robot and get responses within the context of the conversation. This knowledge base was also implemented on the social robot PKD to enhance the conversational abilities of the robot.

7.2 Future Work

The experiments conducted during the presented work can be further extended to include a wide range of users. The experiments can also be extended in the future to be conducted in a healthcare facility to test the proposed framework. The scope of the experiments can be broadened to include the testing of reading of the vitals of the patient. The experiments were conducted on a Baxter robot for the work in the thesis. The next step would be to implement the proposed framework on an ARNA robot.

The framework, proposed in the thesis, can be extended to identify and include other routine tasks that can be performed by a robot to assist the nurses at a healthcare facility. This would help broaden the scope of the ARNA robot. The HMI framework can be further tested to include other smart wearable devices that can be interfaced and be used for recording the data collected by the devices. An alarm system for the nurses can also be included in the framework to incorporate a warning system for immediate assistance for the patient. This framework can also include algorithms to study the vitals from the patient and trigger an early warning system to help achieve proper healthcare and in-time diagnosis for the patient.

The interface used in this work is a tablet but this can be improved upon by using a custom-made interface to further enhance the experience of the user. Further experiments can be conducted to test the performance of various interfaces and their usability in a hospital environment.

The speech interface is based on the knowledge base created from Wikipedia. This can be further enhanced to include web based search engines such as Google and Bing and to be able to extract information from the results. This would be helpful in extracting updated information as well as to include a wide scope of requests such as questions about weather or about the score of a game.

These above-mentioned future enhancements would be able to further enhance the ARNA robot's performance and assist the nurses in a better way.

REFERENCES

- [1] Robin Murphy, *Introduction to AI robotics*, MIT press, 2000.
- [2] Nicholas Roy, Gregory Baltus, Dieter Fox, Francine Gemperle, Jennifer Goetz, Tad Hirsch, Dimitris Margaritis, Michael Montemerlo, Joelle Pineau, Jamie Schulte, et al., “Towards personal service robots for the elderly,” in *Workshop on Interactive Robots and Entertainment (WIRE 2000)*, 2000, vol. 25, p. 184.
- [3] Siddhartha S Srinivasa, Dave Ferguson, Casey J Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garratt Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe, “HERB: a home exploring robotic butler,” *Autonomous Robots*, vol. 28, no. 1, pp. 5–20, 2010.
- [4] Birgit Graf, Ulrich Reiser, Martin Hägele, Kathrin Mauz, and Peter Klein, “Robotic home assistant care-o-bot® 3-product vision and innovation platform,” in *2009 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*. IEEE, 2009, pp. 139–144.
- [5] Toshiharu Mukai, Masaki Onishi, Tadashi Odashima, Shinya Hirano, and Zhiwei Luo, “Development of the tactile sensor system of a human-interactive robot RI-MAN,” *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 505–512, 2008.
- [6] SHIGETOSHI Shiotani, T Tomonaka, K Kemmotsu, S Asano, K Oonishi, and RY-OUTA Hiura, “World’s first full-fledged communication robot” Wakamaru” capable of living with family and supporting persons,” *Mitsubishi Juko Giho*, vol. 43, no. 1, pp. 44–45, 2006.
- [7] Microsoft Kinect, “Kinect for windows sensor components and specifications,” 2017, [Online; accessed 7-July-2017].
- [8] Microsoft Kinect, “Kinect for windows sensor,” 2017, [Online; accessed 7-July-2017].

- [9] Sven Cremer, Kris Doelling, Cody L Lundberg, Mike McNair, Jeongsik Shin, and Dan Popa, “Application requirements for robotic nursing assistants in hospital environments,” in *SPIE Commercial+ Scientific Sensing and Imaging*. International Society for Optics and Photonics, 2016, pp. 98590E–98590E.
- [10] David Feil-Seifer and Maja J Mataric, “Defining socially assistive robotics,” in *ICORR 2005 9th International Conference on Rehabilitation Robotics*. IEEE, 2005, pp. 465–468.
- [11] Kyong Il Kang, Sanford Freedman, Maja J Mataric, Mark J Cunningham, and Becky Lopez, “A hands-off physical therapy assistance robot for cardiac patients,” in *Rehabilitation Robotics, 2005. ICORR 2005. 9th International Conference on*. IEEE, 2005, pp. 337–340.
- [12] Rachid Alami, Alin Albu-Schäffer, Antonio Bicchi, Rainer Bischoff, Raja Chatila, Alessandro De Luca, Agostino De Santis, Georges Giralt, Jérémie Guiochet, Gerd Hirzinger, et al., “Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges,” in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006, pp. 1–16.
- [13] Jeremy A Marvel and Rick Norcross, “Implementing speed and separation monitoring in collaborative robot workcells,” *Robotics and computer-integrated manufacturing*, vol. 44, pp. 144–155, 2017.
- [14] GS Virk, S Moon, and R Gelin, “ISO standards for service robots,” in *Proc. 11th Int. workshop on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, 2008, pp. 133–138.
- [15] Sven Cremer, Fahad Mirza, Yathartha Tuladhar, Rommel Alonzo, Anthony Hingeley, and Dan O Popa, “Investigation of human-robot interface performance in household environments,” in *SPIE Commercial+ Scientific Sensing and Imaging*. International Society for Optics and Photonics, 2016, pp. 985904–985904.
- [16] Rommel Alonzo, Sven Cremer, Fahad Mirza, Sandesh Gowda, Larry Mastromoro, and Dan O Popa, “Multi-modal sensor and HMI integration with applications in personal

- robotics,” in *SPIE Sensing Technology+ Applications*. International Society for Optics and Photonics, 2015, pp. 949409–949409.
- [17] Wikipedia, “Chatbot - Wikipedia, The Free Encyclopedia,” 2017, [Online; accessed 7-July-2017].
- [18] Namita Mhatre, Karan Motani, Maitri Shah, and Swati Mali, “Donna Interactive Chatbot acting as a Personal Assistant,” *International Journal of Computer Applications*, vol. 140, no. 10, 2016.
- [19] Michelle Denise Leonhardt, Liane Tarouco, Rosa Maria Vicari, Elder Rizzon Santos, and Michele dos Santos da Silva, “Using chatbots for network management training through problem-based oriented education,” in *Advanced Learning Technologies, 2007. ICAALT 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 845–847.
- [20] Agnese Augello, Antonella Santangelo, Salvatore Sorce, Giovanni Pilato, Antonio Gentile, Alessandro Genco, and Salvatore Gaglio, “A multimodal interaction guide for pervasive services access,” in *Pervasive Services, IEEE International Conference on*. IEEE, 2007, pp. 250–256.
- [21] Fernando A Mikic, Juan C Burguillo, Martín Llamas, Daniel A Rodríguez, and Eduardo Rodríguez, “CHARLIE: An AIML-based Chatterbot which Works as an Interface among INES and Humans,” in *EAAEEIE Annual Conference, 2009*. IEEE, 2009, pp. 1–6.
- [22] De Pietro Orlando and Frontera Giovanni, “An integrated system, with natural language management, for the monitoring activities in e-learning environments,” in *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*. IEEE, 2008, pp. 965–970.
- [23] Yu Wu, Gongxiao Wang, Weisheng Li, and Zhijun Li, “Automatic chatbot knowledge acquisition from online forum via rough set and ensemble learning,” in *Network and Parallel Computing, 2008. NPC 2008. IFIP International Conference on*. IEEE, 2008, pp. 242–246.
- [24] Agnese Augello, Giovanni Pilato, Giorgio Vassallo, and Salvatore Gaglio, “A semantic layer on semi-structured data sources for intuitive chatbots,” in *CISIS’09. International*

- Conference on Complex, Intelligent and Software Intensive Systems*. IEEE, 2009, pp. 760–765.
- [25] Wikipedia, “Wikipedia - Wikipedia, The Free Encyclopedia,” 2017, [Online; accessed 7-July-2017].
- [26] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman, “Indexing by latent semantic analysis,” *Journal of the American society for information science*, vol. 41, no. 6, pp. 391, 1990.
- [27] Robert P Schumaker, Ying Liu, Mark Ginsburg, and Hsinchun Chen, “Evaluating mass knowledge acquisition using the ALICE chatterbot: The AZ-ALICE dialog system,” *International Journal of Human-Computer Studies*, vol. 64, no. 11, pp. 1132–1140, 2006.
- [28] Elizabeth Broadbent, Rebecca Stafford, and Bruce MacDonald, “Acceptance of health-care robots for the older population: review and future directions,” *International journal of social robotics*, vol. 1, no. 4, pp. 319–330, 2009.
- [29] Rafael V Aroca, Antônio Péricles, BS de Oliveira, L Marcos, and G Gonçalves, “Towards smarter robots with smartphones,” in *5th workshop in applied robotics and automation, Robocontrol*. sn, 2012, pp. 1–6.
- [30] Juan C Yepes, Juan J Yepes, Jose R Martinez, and Vera Z Pérez, “Implementation of an Android based teleoperation application for controlling a KUKA-KR6 robot by using sensor fusion,” in *2013 Pan American Health Care Exchanges (PAHCE)*. IEEE, 2013, pp. 1–5.
- [31] Tatiana Alexenko, Megan Biondo, Deya Banisakher, and Marjorie Skubic, “Android-based speech processing for eldercare robotics,” in *Proceedings of the companion publication of the 2013 international conference on Intelligent user interfaces companion*. ACM, 2013, pp. 87–88.
- [32] Kris Doelling, Jeongsik Shin, and Dan O Popa, “Service robotics for the home: a state of the art review,” in *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2014, p. 35.

- [33] Damon Kohler and Ken Conley, “rosjava—An implementation of ROS in pure Java with Android support,” 2011.
- [34] Richard Wallace, “The elements of AIML style,” *Alice AI Foundation*, 2003.
- [35] Thurid Vogt, Elisabeth André, and Johannes Wagner, “Automatic recognition of emotions from speech: a review of the literature and recommendations for practical realisation,” in *Affect and emotion in human-computer interaction*, pp. 75–91. Springer, 2008.
- [36] Agnese Augello, Gaetano Saccone, Salvatore Gaglio, and Giovanni Pilato, “Humorist bot: Bringing computational humour in a chat-bot system,” in *CISIS 2008 International Conference on Complex, Intelligent and Software Intensive Systems*. IEEE, 2008, pp. 703–708.
- [37] Susan T Dumais, George W Furnas, Thomas K Landauer, Scott Deerwester, and Richard Harshman, “Using latent semantic analysis to improve access to textual information,” in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 1988, pp. 281–285.
- [38] George W Furnas, Scott Deerwester, Susan T Dumais, Thomas K Landauer, Richard A Harshman, Lynn A Streeter, and Karen E Lochbaum, “Information retrieval using a singular value decomposition model of latent semantic structure,” in *Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1988, pp. 465–480.
- [39] Andrian Marcus and Jonathan I Maletic, “Recovering documentation-to-source-code traceability links using latent semantic indexing,” in *Software Engineering, 2003. Proceedings. 25th International Conference on*. IEEE, 2003, pp. 125–135.
- [40] David Hanson, “Exploring the aesthetic range for humanoid robots,” in *Proceedings of the ICCS/CogSci-2006 long symposium: Toward social mechanisms of android science*, 2006, pp. 39–42.
- [41] Jun-Ho Oh, David Hanson, Won-Sup Kim, Young Han, Jung-Yup Kim, and Ill-Woo Park, “Design of android type humanoid robot Albert HUBO,” in *Intelligent Robots*

- and Systems, 2006 IEEE/RSJ International Conference on.* IEEE, 2006, pp. 1428–1433.
- [42] Takuya Hashimoto, Sachio Hitramatsu, Toshiaki Tsuji, and Hiroshi Kobayashi, “Development of the face robot saya for rich facial expressions,” in *SICE-ICASE, 2006. International Joint Conference.* IEEE, 2006, pp. 5423–5428.
- [43] Hanson Robotics, “ABOUT HANSON ROBOTICS,” 2017, [Online; accessed 7-July-2017].
- [44] RethinkRobotics, “Baxter arms,” 2017, [Online; accessed 7-July-2017].
- [45] RethinkRobotics, “Baxter camera,” 2017, [Online; accessed 7-July-2017].
- [46] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software.* Kobe, 2009, vol. 3, p. 5.
- [47] Nathan Koenig and Andrew Howard, “Gazebo-3d multiple robot simulator with dynamics,” 2006.
- [48] Jorge J Villalón and Rafael A Calvo, “A decoupled architecture for scalability in text mining applications.,” *J. UCS*, vol. 19, no. 3, pp. 406–427, 2013.

CURRICULUM VITAE

NAME: Ankita Sahu

ADDRESS: Computer Engineering & Computer Science Department
Speed School of Engineering
University of Louisville
Louisville, KY 40292

DOB: 27th August 1988

EDUCATION:

M.S., Computer Science & Engineering

August 2017

University of Louisville, Louisville, Kentucky

B.Tech., Biotechnology Engineering

June 2010

Visvesvaraya Technological University, Bangalore, India

CONFERENCE PUBLICATIONS:

1. Sumit Kumar Das, **Ankita Sahu** and Dan O. Popa, “*Mobile app for human-interaction with sitter robots*”, in Proc. SPIE. 10216, Smart Biomedical and Physiological Sensor Technology XIV, 102160D. (May 16, 2017) doi: 10.1117/12.2262792.