VERSION 1.0

11.25.2018

PRESENTED BY: CASEY CHEN, HAILEY DHANENS, MATTHEW HARKER, ANGIE QUACH, DEREK VAUGHAN

ARIGATO
ELLENSBURG, WA

CONTENTS

SECTION 1: INTRODUCTION

1.1 PROBLEM

Our mission is to showcase the capabilities of the Aldebaran NAO robots recently acquired by the Central Washington University computer science department robotics lab with human-robot interaction. The client specifically wants to showcase the robot's ability to follow basic commands, recognize voice commands, and recognize human emotions.

1.2 SCOPE

The Aldebaran NAO is pre-packaged with a vast library of Application Programming Interfaces (APIs) consisting of libraries of functions that will help us achieve our goals mentioned in Section 1.1. Our team will create several modules for the NAO robots called NAOqi's that will perform the tasks desired by the client. These tasks will include interacting with the user via verbal communication and physical responses. The goal of these responses is to showcase the abilities of the NAO robot, so it is important to make sure NAO has a multitude of responses and functions, as the NAO robot has extensive capabilities.

Many of NAO's features make use of its built-in high definition cameras and sound processors, which allow it to take in input through a variety of methods and produce output. The NAO does not require a constant connection to the internet, however some features will be limited when "off-the-grid"; such as when the NAO needs to fetch data or perform certain calculations. We have the ability to upload programs and features to the NAO's memory unit – meaning it will not need to be connected to a computer or the internet at all times.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

ALDEBARAN
A French company focused on robotics that manufactures the NAO robot. It was formed in 2009, and based in Tokyo, Japan.

API
Application Program Interface.

CHOREGRAPHE

A multi-platform desktop application that allows users to create animations and behaviors for the NAO and test them in both simulated and real environments. It also allows users to monitor the NAO's visual and audio sensors.

CLIENT
Dr. Szilard Vajda, assistant professor, of the Central Washington University Computer Science department, the requester of this project

ETHERNET
A common form of network cable. It allows a connected device to join a local area network (LAN) in order to connect to and browse the internet.

LIBRARY
A collection of well-defined resources and implementations of behavior, written for/in a particular programming language for use by other developers to simplify and speed up development for a system.

NAO
A programmable, humanoid robot designed by Aldebaran Robots.

NAOQI
A Linux-based operating system stored in the robot's memory at all times; used for running and controlling features and programs.

SEE
The NAO robots cannot "see" in a physical sense but has cameras that it can use to record images to identify its surroundings.

SENSOR
Measures the robot's configurations, conditions, and its environment and sends such information to the robot for processing.

SOFTWARE
A set of computer instructions used to obtain input, and then manipulate that input in order to generate relevant output in terms of function and performance as specified by the user.

SOURCE
A university-wide event in May 2019 showcasing all disciplines of research, scholarship, and creative activities by students, faculty, and staff.

USER
A person who will interact with and make use of the NAO's various capabilities.

1.4 REFERENCES

[1] Guide to making an SRS: http://www.cse.msu.edu/~cse870/IEEEXplore-SRS-template.pdf
[2] NAO Lab Documentation/Info: https://team.inria.fr/perception/demos/naolab/
[3] NAO, NAOqi, Choregraph Documentation: http://doc.aldebaran.com/
[4] NAO, Technical Guide: http://doc.aldebaran.com/2-1/family/index.html
[5] Engineering NYU, Intro to Robotics
http://engineering.nyu.edu/mechatronics/smart/pdf/Intro2Robotics.pdf

1.5 OVERVIEW

The purpose of this document is to describe the requirements of AriGato's Aldebaran NAO project. Section 2 provides an overall description of the project and its requirements, including the hardware, users and individual functionalities. This section is followed by a requirements overview (section 3) that describes more in-depth the required functionalities of the NAO robot, as well as a question and answer section. Section 4 then analyses the validity, feasibility, and consistency of these requirements. Finally, section 5 prioritizes these requirements.

SECTION 2: OVERALL DESCRIPTION

This project will be independent and self-contained. It will not be a part of a larger system.

2.1 PRODUCT PERSPECTIVE & FUNCTIONALITY

2.1.1 HARDWARE

This project will require the use of an Aldebaran NAO robot. Our team will have access to an NAO robot in the Central Washington University computer science department robotics lab in the Samuelson Building, as well as a computer that can run the Choregraphe Software Development Kit. Choregraphe is fully functional on Windows and available on Linux with limited support. Additionally, a computer equipped with an Ethernet port as well as an Ethernet cable are necessary for computer-robot interaction.

2.1.2 USER INTERFACES (CONTROL METHODS)

Our software has two possible modes of operation: The first by using a computer as a control terminal. In this mode, the NAO will be connected via Ethernet cable to a computer running the Choregraphe. Choregraphe is a program that allows users to send premade or custom-made sets of instructions to the NAO, which will then be acted upon. This mode is limited however, as the robot can only go as far as the attached Ethernet will allow (basically, the robot must be in very close proximity to the control terminal computer). This mode is mainly for testing purposes.

The second mode, and main focus of our software project, is directly uploading our software to the NAO robot's memory unit located in its head. This will allow the NAO to operate with full functionality while completely disconnected from a computer, i.e., an Ethernet cord will not need to be attached to the robot for it to receive instructions. The NAO is capable of this

thanks to its Wi-Fi connection capabilities. The users will be able to interact with the robot with touch, motion, and spoken commands using the multitude of sensors located around NAO's body.

2.2 SUGGESTED FUNCTIONALITIES

- Make the robot solve written equations
- Give the robot ability to read and point out simple objects
- Give the robot an understanding of American Sign Language (ASL) or simple hand gestures
    - Difficult because it does not already have an API
    - NAO API has waving detection
    - Rock Paper Scissors
    - Secret handshake
- Human/Robot game
    - Like "Bop It"
    - Human is given instructions to perform within a certain amount of time
- Make robot act more human-like
    - Casual standing position
    - Reacts to stimulus with body language

2.3 USER CHARACTERISTICS

There will be two classes of users that will interact with the NAO unit: regular users, and administrators (who also act as demonstrators).

The NAO robot will be interacting with a wide range of people that classify as "regular users," such as computer science students, department faculty and staff, the general university population, university visitors, and more. The Users will be able to have differing levels of experience. As previously mentioned, we envision the NAO to be used as a demonstration tool for any person that might be interested in the computer sciences, such as primary school students. Regular users will interact with the NAO by giving it verbal commands and specific gestures or movements to invoke a corresponding response.

Administrators are people who will manage the NAO unit after our software team has delivered the final product. These people will manage the storage and maintenance of the NAO, as well as plugging the robot into control terminals, and charging it. They will also be responsible for keeping the NAO updated to any new firmware changes from Aldebaran, and possibly adding new functionality to our software in the future, if desired. Finally, administrators will typically be the ones to demonstrate the NAO's full range of capabilities to anyone interested in the robot.

2.4 CONSTRAINTS AND LIMITATIONS

The NAO model in use contains 8GB of onboard memory, meaning all code and additional data, such as stored images, will have to be able to be stored within this memory. While it is unlikely that this constraint will become a problem, it is something that should be kept in mind.

Another unfortunate constraint of the NAO robot is the overall quality of its various sensors around its body; including, but not limited to: its microphones, cameras, and gyroscopes. Although the NAO comes packed in with HD cameras in its eyes, it occasionally has difficulties locating a target (e.g., a user, or an item in front of it). Additionally, the sensors are not always accurate because a robot's environments are always changing, so it will be taking in a lot of information at once.

Furthermore, through our team's independent study of the NAO's capabilities we have noticed that the unit's microphones require commands to be given loudly, slowly, and with extremely clear pronunciation in order for the NAO to correctly parse the commands and give appropriate responses. It is also difficult for the microphones to pick up specific commands if it's in a crowded room with many different voices talking. Despite these limitations, our team will work diligently to make our NAO software perform quickly and accurately. Furthermore, the NAO should always be kept connected to the internet to make use of all the offered features. Though the NAO is functional without an internet connection, some commands will potentially require the NAO to make use of an internet connection to provide a response. This should not be an issue, as the NAO is primarily kept within university premises.

Given the cost of the robot, we are also mindful of not programming the robot to do strenuous physical activity. If something were to go wrong, the robot may end up walking off a table, or falling onto a sharp object. The price tag is too inhibitive for more than simple actions like walking around or picking up objects. The robot's arms have many mechanical joints which move along three axes. When programming an arm joint to specifically move, it could damage the hardware if the arm tries to move in a way it physically cannot. The same goes for NAO's leg and finger joints.

A small note should be made that the NAO robot has about a 90-minute active-use battery life, meaning that future users will only be able to use the NAO unplugged for around an hour and a half before it must be charged again. This constraint should not affect our development, as we will always be near charging ports and control terminal computers.

2.5 ASSUMPTIONS & DEPENDENCIES

The first assumption we've made about the NAO is that it will typically, if not always, be kept on university premises, where its status will be resting when not in use, and will always be in range of an internet connection. In the case that a future administrator of the NAO would like to take the unit out of the university network, it will need to be configured to the new location's internet settings.

The NAO robot can be programmed to do tasks such as using web APIs, but those functions would be a small portion of what it will be able to do. The assumption being made is that it will have access to the internet while being demonstrated. Not having any internet access would prevent it from being able to perform tasks using web API's.

Another assumption is that users will expect only the listed set of commands to be the set of functionalities the robot has (i.e., there is a finite list of functionality). Of course, there are a few small built-in commands that each NAO unit has (such as "What is your IP address?"), aside from these small built-in functions, the only things the NAO will be capable of doing is what we program it to do (this will be gone over in extensive detail in section 3.1).

Finally, we are assuming that the hardware of the NAO unit will work properly at all times. However, as mentioned in section 2.3, the video and audio processing capabilities of the NAO do occasionally limit the robot from understanding every command correctly. We will be working around these potential technical limitations, and assume that the sensors will all work properly, 100% of the time. To clarify, no work will be done to improve the sensors of the NAO, as that is out of the scope of our software-focused project.

2.6 HIGH LEVEL DESIGN

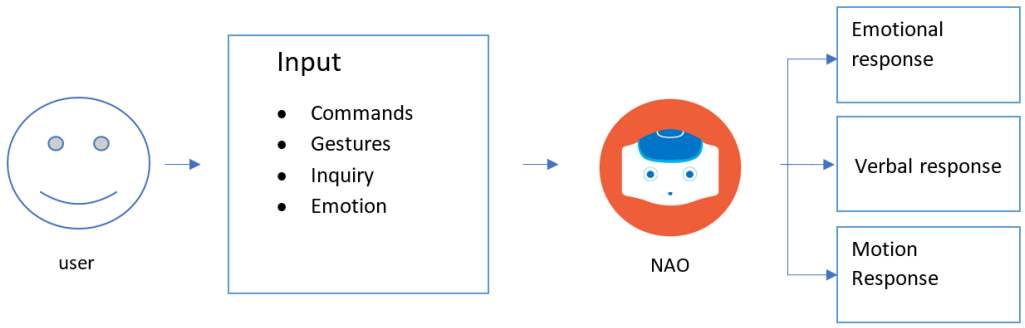

FIGURE 1 - USER-ROBOT INTERACTION

The user will input to the robot using a variety of methods, like verbal commands, questions, gestures, and emotion. NAO will process and respond to these inputs with emotional responses, verbal responses and movement, depending on the input.

SECTION 3: REQUIREMENTS OVERVIEW

3.1: REQUIREMENTS CHECKLIST

NAO should:

- Be able to quickly recognize and act upon basic commands and gestures from a specific finite list (full list to be determined within a month).
- Be able to quickly answer all questions and inquiries from a specific finite list (full list to be determined within a month).
- Be able to demonstrate the movement of each appendage (arms, fingers, legs, etc.) when reacting to certain commands or gestures.
- Exhibit some form of emotional response based on the way a user has given it a command (e.g., shouted angrily, laughing).
- Be able to recognize when a user has said “foul” language, and react by telling the user to watch their language, or some similar phrase.
- Come provided with extensive documentation on the how’s, what’s, and why’s of the software system that has been programmed for it.

- Come provided with some form of simplified user manual to allow any first-time users to be able to easily pick up and fully interact with the NAO robot (while it utilizes our software).

3.2: FUNCTIONAL REQUIREMENTS

- Be able to quickly recognize and act upon basic commands and gestures from a specific finite list (full list to be determined within a month). **[Requirement ID: FR-1]**
- Be able to quickly answer all questions and inquiries from a specific finite list (full list to be determined within a month). **[Requirement ID: FR-2]**
- Be able to demonstrate the movement of each appendage (arms, fingers, legs, etc.) when reacting to certain commands or gestures. **[Requirement ID: FR-3]**
- Be able to recognize when a user has said "foul" language, and react by telling the user to watch their language, or some similar phrase. **[Requirement ID: FR-4]**

3.3: NON-FUNCTIONAL REQUIREMENTS

- Exhibit some form of emotional response based on the way a user has given it a command (e.g., shouted angrily, laughing). **[Requirement ID: NFR-1]**
- Come provided with extensive documentation on the how's, what's, and why's of the software system that has been programmed for it. **[Requirement ID: NFR-2]**
- Come provided with some form of simplified user manual to allow any first-time users to be able to easily pick up and fully interact with the NAO robot (while it utilizes our software). **[Requirement ID: NFR-3]**

3.4: REQUIREMENTS Q & A

What will the system do?

- The goal of this project is to give the robot multiple functionalities to be displayed.

Should there be different modes of operation?

- Different functionalities should be available, as described in section 2.2, "Suggested Functionalities".

What format should be considered for the input/output?

- Input for the robot will be verbal commands, touch, motion of the user and equations written on paper.
- Output will be verbal responses and movements

Should we code responses to verbal commands?

- The robot should be able to respond to verbal commands as required by the operation it is performing.

Should the robot be able to recognize specific people?

- Facial recognition is not required by this project.

How quickly should NAO react to a change in facial expression?

- Facial expression recognition is not currently one of the required functionalities.

How many emotions and what emotions should NAO be able to respond to?

- NAO can recognize smiles in the API as well as different emotions in voices.

How should the NAO react? Verbally? With body language? Use of the NAO eyes? A mixture of these?

- NAO should be able to react verbally and with body language/movements.

Who might be changing our code in the future?

- Future CS 481 students, as well as robotics students, may be viewing and editing our code in the future.

How easy should we make it to add our functionality to a different kind of robot? Will that even be possible with the NAO API?

- Moving NAO functionality to other Aldebaran robots could be a challenge as the PEPPER has different functionalities than NAO, as it doesn't have legs and has a screen on its chest.

How many NAO units will we have access to?

- The team will have access to a single NAO robot in the CWU computer science robotics lab

When will we be able to have access to the NAOs?

- The team will be able to access the robotics lab during Samuelson's open hours if the robotics lab is not already in use (M-F until 8 PM) - the room is currently only reserved from 9-11 AM Monday to Thursday for Fall quarter 2018

What documentation is required?

- Create a poster describing NAO's most exciting features, as well as write a manual describing our added features

In what environments will the NAO be used?

- NAO will be used in the robotics lab setting, as well as possibly being showcased at the school's SOURCE presentation in 2019.

What programming languages will be used?

- Python 2.7

Who are the users?

- Users will be whoever enters the robotics lab. This could be prospective students, current students, and the professors, as well as the people attending source.

Where can information on the NAO API be found?

- Information on the NAO API can easily be found by looking up "Aldebaran NAO" online and finding the ROBOTS documentation.

Can NAO identify the face, and does NAO have any facial mapping algorithms already equipped?

- NAO has an API that locates the user's face and maps out its important features.

SECTION 4: REQUIREMENTS ANALYSIS

4.1 VALIDITY CHECK

The purpose of all the suggested requirements is to showcase the abilities of the robot, as specified by the client. Following basic commands showcases the robot's ability to understand language as well as its ability to follow through commands. Reacting differently to an angry, shouted command than a normal spoken command will show its ability to recognize emotion in humans. Identifying the programmers and learning names will show off NAO's facial mapping and facial recognition abilities.

Our software's requirements are **valid.**

4.2 FEASIBILITY CHECK

Our team of software engineers have extensive experience in all tools and technologies mentioned in this document. Each of us has already developed and ran small test programs on the NAO unit, and performed extensive research on the different libraries and capabilities of the NAO. We can confirm that there will be no conflicts in the different tools that we will make use of (Python, Choregraphe, etc.) and put together. Aldebaran's provided APIs will allow a very timely development period, as functions such as "move right hand upwards" for example are already written and tested for us. It will be up to our team to make use of and combine these provided functionalities to make an overall quality software package.

Some of the suggested functionalities are not feasible in the amount of time that we have. Learning American Sign Language (ASL) was a suggestion, however, NAO does not recognize finger positions, and writing a function to do this would not take an incredible amount of time and study but is also out of the scope of the team's current skills.

It is currently unclear how the NAO API's facial recognition function works; however it does have facial mapping. It may be difficult, but facial recognition is a distinct possibility

Our software's requirements are **feasible.**

4.3 CONSISTENCY CHECK

Thorough analysis of our requirements checklist has not shown any conflicts of consistency. The Choregraphe software is capable of taking in as input all of the different languages and libraries discussed in this document and allows us to easily test new code in a safe environment. The NAO robot is capable of taking on different "states", such as a "waiting state", where it will sit idly and wait for some command to be given, or an "active state" where the NAO is currently performing an action. This will allow all of our functions and methods to concurrently exist in the robot's memory and be acted upon when necessary (i.e., when a command is given). There will be no consistency conflicts in our project.

Our software's requirements are **consistent.**

4.4 PROJECT TIMELINE

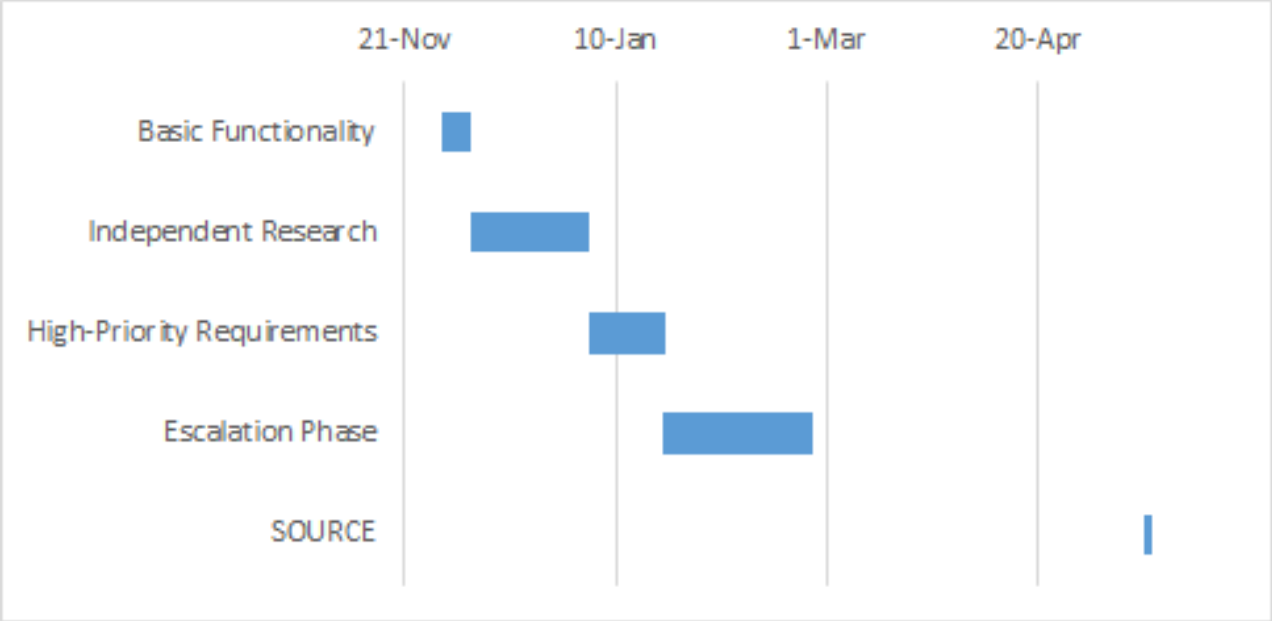

FIGURE 2 - PROJECT TIMETABLE

The team will begin the project learning how to program basic functionality, until the winter break, in which they will begin independent research. January 3rd, when the break ends, the team will begin working on the high priority requirements. The team will divide into smaller groups of one to two, each working on a set of modules, and meeting weekly to test and discuss projects. When all the high priority modules have been completed, the escalation phase will begin.

During the escalation phase, team members will be in larger groups of 2 to 3, as the lower priority modules are more complex, and will require more work. Again, the teams will meet once a week to test modules and discuss if the groups should take on more modules. At the end of February, the final testing will begin, and the project will be presented to the client. Finally, May 15th to May 16th, the team will present the NAO robot at CWU The Symposium Of University Research and Creative Expression (SOURCE).

SECTION 5: PRIORITIZATION OF REQUIREMENTS

Based on our conversations with the client, the main requirement we should focus on is getting the robot to be controlled independent of the Choregraphe software. Specifically, the

priority is to get the NAO robot to be able to understand human commands and interpret it both to perform the action and to analyze *how* the person is speaking to the robot.

Goals that will absolutely be met:

- Listen to simple commands and be able to interpret and act upon them.
- Respond in an appropriate way depending on the emotion behind the speaker's voice.
    - For example, refuse a request if it was made in an angry way.
- Be able to pick up a red ball (the NAO is limited to red ball recognition for pick up).

Goals that we will strive to meet:

- Read/solve written equations.
- Dance to music.

Goals that are possible, or "Stretch-Goals":

- Be able to recognize when a user has said "foul" language and react.
- Extra features, i.e., "Bop It" game.