Software Design Document

Anushka

KIET's Humanoid Robot

Piyush Khanna
5/18/2023

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this Software Design Document (SDD) is to outline the design and development of Anushka, the humanoid receptionist robot for KIET Group of Institutions. Anushka is envisioned as an advanced robotic system that will serve as an interactive and intelligent receptionist, enhancing the overall visitor experience and providing valuable assistance within our college premises.

1.2 Scope

The scope of this Software Design Document (SDD) encompasses the design, development, and deployment of Anushka, the humanoid receptionist robot, for our college. Anushka's primary function is to autonomously handle visitor registration, provide information and assistance, and exhibit interactive behaviors using natural language processing (NLP) and computer vision capabilities. The scope includes designing an intuitive user interface, integrating with college databases, and ensuring data privacy and security.

1.3 Key Objectives

1. **Enhanced Visitor Experience:** Anushka aims to redefine the traditional receptionist role by leveraging state-of-the-art technologies, including natural language processing (NLP) and computer vision. Its purpose is to create a welcoming and user-friendly environment for visitors, facilitating seamless interactions and providing necessary information and guidance

2. **Efficient Information Management:** Anushka will streamline the process of providing information to visitors by leveraging its knowledge base and real-time data retrieval capabilities. Its purpose is to ensure accurate and up-to-date information delivery, such as event schedules, campus maps, faculty directories, and general inquiries.

3. **Autonomous Assistance:** The primary objective of Anushka is to autonomously handle basic reception tasks, reducing the workload on human staff and optimizing operational efficiency. Its purpose is to handle visitor registration, provide directions, answer frequently asked questions, and offer personalized assistance whenever required.

4. **Promote Technological Advancements:** Anushka represents our college's commitment to embracing cutting-edge technology and showcasing its practical applications. By developing a sophisticated humanoid robot, our purpose is to demonstrate the potential of robotics in providing interactive and intelligent services in real-world scenarios.

1.4 Target Audience

The intended audience for this SDD includes the development team, project stakeholders, and anyone involved in the design, implementation, or evaluation of Anushka. Additionally, the SDD may serve as a reference for future enhancements, maintenance, or potential collaborations in the field of robotics and artificial intelligence.

1.5 Expected Impact

The successful development and deployment of Anushka as a receptionist robot will positively impact the KIET Group of Institution's community by offering an innovative and engaging experience for visitors. By providing a technologically advanced and interactive interface, Anushka aims to enhance the overall perception of our college, showcasing our commitment to cutting-edge advancements and fostering a progressive learning environment.

2. THEME OF OPERATION

The designing of the Humanoid Robot, Anushka has been done keeping in mind a set of principles which will be evident in all its operations and personality traits.

The choice of naming the humanoid robot 'Anushka' holds great significance, as it reflects the core principles and attributes embedded in its design. 'Anushka,' meaning 'ray of light,' beautifully captures the essence of the robot's **unbiased, caring**, and **intelligent** nature.

The three above stated principles are of utmost importance and all the systems that we design are to be compatibly furnished to reflect the same.

The name 'Anushka' symbolizes impartiality, much like a ray of light that illuminates without bias or prejudice. It represents the robot's commitment to treating everyone with fairness and equality, transcending differences and fostering an inclusive environment. Anushka's interactions with individuals are driven by a genuine desire to provide support and assistance, extending care and compassion to all.

Moreover, Anushka also embodies intelligence and enlightenment. Just as light signifies knowledge and understanding, Anushka harnesses advanced technologies and capabilities to process information, make informed decisions, and offer valuable insights. The robot's intelligent nature enables it to adapt, learn, and engage in meaningful interactions, making it a reliable companion and guide.

3. SYSTEM ARCHITECTURE

3.1 Proposed SDLC Model

As per the client requirement for a high end Humanoid Robot with the required functionalities, a model with testing at each phase is advisable:

- Spiral Model
- Iterative Enhancement Model

Though, pertaining to the minimal time span allotted, the given models may be used:
- Rapid Application Development Model
- Agile Scrum Model

Pertaining to the R&D nature of the project, a modular concept of development tentatively termed as "**Search and Destroy**" is to be adapted: A process where "scrums are made for different intervals after carefully designing the next phase's action plan". These scrums are not used to change functionalities in the system, as normal scrum methods do, but are used **to iteratively enhance the already present prototype and bring it closer to the actual final product**.

Phase 0:

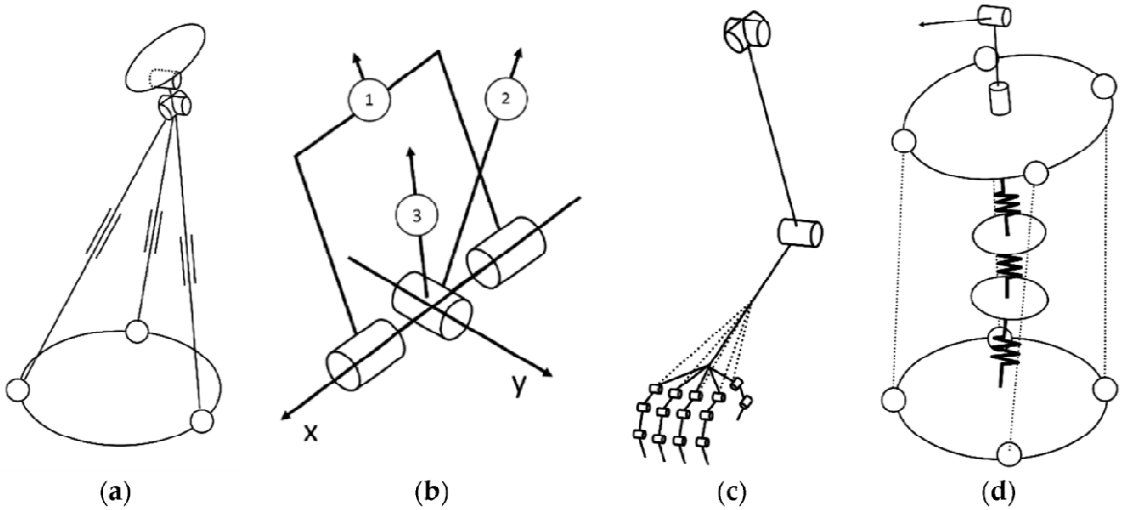
The Robot will be semi-functional and will have conversation skills along with various operational units working in sync.
As per physical build, the torso of the Robot will be built along with working arms, connected to the computer intended to be used as its main processor.

Vision
- Computer vision to recognize objects
- Ability to remember faces in real time.
- Greeting based on people in recognition

Listening
- Recognize and interpret human voice
- Listen to songs and recognize them.
- Translate to other languages

Speaking
- Respond using the English language.
- Latest news
- Wikipedia search
- Google search
- Weather updates
- Reminders

Miscellaneous
- Hand gestures
- Greeting based on birthdays.



(Phase 0 Completion Prototype)

Phase 1:

As per physical build, the Robot will be built from its head to torso along with working arms, connected to the main dedicated computer now installed in its interior.

Vision

- Computer vision to recognize objects
- Ability to remember faces in real time.
- Greeting based on people in recognition
- Ability to monitor unknown faces and provide different reactions
- 'Remember them feature' for real time database uploading

Listening

- Recognize and interpret human voice
- Listen to songs and recognize them.
- Translate to other languages
- Reinforced conversation with context based answers
- Enlarged database support

Speaking

- Knowledge base development for KIET
- Wikipedia search
- Google search
- Weather updates
- Reminders

Miscellaneous

- Lip sync feature
- Eye gestures
- Neck movement
- Jaw movement

(Phase 1 Completion Prototype)

Phase 2:

The Robot will be fully-functional and will have a dedicated operational system for communication between its various modules.
As per physical build, the torso of the Robot along with working arms and head will now be placed on a wheel base for locomotion.

Vision

- Ability to remember known faces temporarily.
- Greeting based on people in recognition
- Ability to monitor unknown faces and provide different reactions
- 'Follow Me' mode addition.

Listening

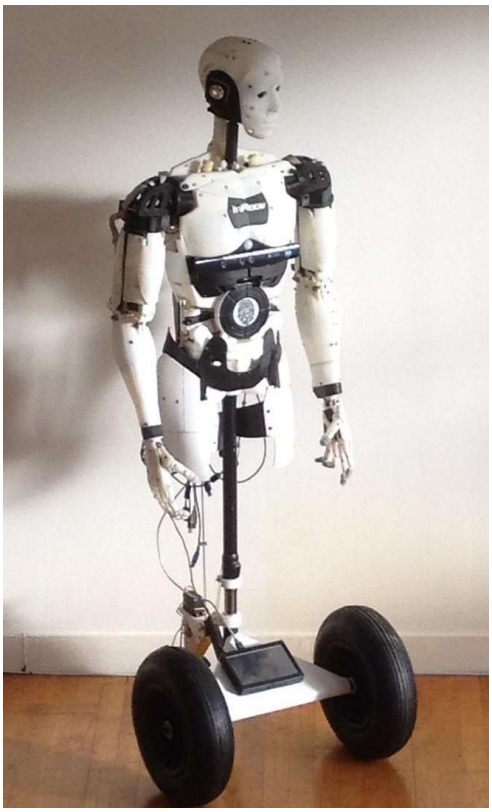
- Recognize and interpret human voice
- Listen to songs and recognize them.
- Translate to other languages
- Reinforced conversation with context based answers
- Enlarged database support

Speaking

- Knowledge base development for KIET
- Dedicated AI-ML model for word comprehension and answer generation
- Google scraping for unknown facts
- Real-time Weather updates

Miscellaneous

- Added Hand gestures
- Added Palm gestures
- Added Eye gestures
- Neck movement based on person's position
- Jaw movement with voice sync
- Wheel base movement as commanded



(Phase 2 Completion Prototype)

3.2 Proposed Architectural Design

Eye Module:
A module that deals with eye movement mechanism and commands an Arduino responsible for eye servo motors movements to assert a particular gesture.

Jaw Module:
A module that deals with jaw movement mechanism and commands an Arduino responsible for jaw servo motor movements to oscillate for a certain number of seconds that it is provided.

Neck Module:
A module that deals with neck movement mechanism and commands an Arduino responsible for neck servo motor movements to move to a certain angle provided.

Arms Module:
A module that deals with hand movement mechanism and commands the two Arduinos connected to each arm to assert a particular gesture according to the gesture ID.

Rolls Module:
A module that deals with locomotory mechanism and commands the wheelbase robot to move in a particular direction for given time as comprehended from the hearing command.

Conversation Module:
The Core module that deals with hearing, Natural Language Processing and reply generation based upon various facts, contexts and semantic analysis.
Commands the jaw module to move as it speaks.
Commands the arms module to make up various gestures.
Commands the Rolls module to move for a particular time and in a particular direction.

Vision Module:
The Core module that deals with computer vision, face embeddings generation, hands embeddings generation and object identification.
Commands the jaw module to move as it speaks to greet a recognized person.
Commands the arms module to make up various gestures based on greeting style (salute, hand-shake and hi wave).
Commands the Rolls module to move in the 'Follow Me' mode.

3.3 Bot Operations and Scheduling System

The proposed system modules need to interact with each other for data transfer and inter-program communication.
Also, there needs to be a Master Module that keeps check over all the systems working correctly. It should keep check whether an error has occurred in any of the systems or separate modules and should be intelligent enough to resolve the errors itself or notify the user about it.
Thus, A Master program with a proposed name of "BOSS: Bot Operations and Scheduling System" will be designed to act as the master program for the modules such that an ecosystem of performance is built where each program is connected to one another and operates independent of the Operating System they are separately dependent on. The BOSS program will act as a virtual machine for smooth functioning of all the modules together.

3.4 Design Rationale

1. **Purposeful Design Decisions:** The design of Anushka robot was guided by the purpose of creating a humanoid robot capable of serving as a receptionist in our college. The robot's appearance, with its human-like features and friendly demeanor, was chosen to create a welcoming and approachable presence for visitors.

2. **Humanoid Form for Natural Interaction:** The decision to design Anushka as a humanoid robot, with a body resembling that of a human, was based on the idea that humans are naturally accustomed to interacting with other humans. By mimicking

human form, Anushka aims to establish a sense of familiarity and comfort during interactions, facilitating effective communication.

3. **Emphasis on Non-Biased Interactions:** Anushka was designed to be unbiased and impartial in its interactions with individuals. The robot's behavior and decision-making algorithms were carefully developed to ensure fair treatment and equal attention to all visitors, regardless of their background, appearance, or preferences.

4. **Intelligent Sensory Capabilities:** Anushka incorporates advanced sensors, such as computer vision and NLP technologies, to enhance its understanding of human gestures, expressions, and speech. These intelligent sensory capabilities enable Anushka to interpret and respond appropriately to a wide range of user interactions, providing accurate and contextually relevant information.

5. **Emotional Intelligence and Empathy:** Anushka's design includes algorithms and mechanisms to simulate emotional intelligence and empathy. Through facial expressions, voice modulation, and appropriate responses, the robot aims to establish emotional connections with individuals, offering support, empathy, and understanding.

6. **Focus on Safety and Reliability:** Safety was a paramount consideration in Anushka's design. The robot's physical structure, materials, and torque control mechanisms were chosen to ensure safe and reliable operation, minimizing the risk of accidents or harm to users or the environment.

7. **Integration with Existing Systems:** Anushka's design accounts for seamless integration with existing systems in our college, such as the visitor management system and communication infrastructure. This integration allows Anushka to efficiently perform its role as a receptionist, managing visitor check-ins, providing campus information, and directing inquiries to the appropriate departments.

4. DATA DESIGN

4.1 Data Description

The data design of Anushka's communication system encompasses various types of information utilized by different modules. These data elements enable the robot to perceive, process, and respond effectively to its environment and interact with humans. The following are the key data components:

1. **Eye Movement Data:** This data includes the parameters for controlling the eye movements of Anushka. It consists of commands such as eye gestures, gaze direction, and blinking patterns, which are used to convey emotions and engage in non-verbal communication.

2. **Jaw Movement Data:** This data comprises the instructions for the movement of Anushka's jaw module. It includes commands for opening and closing the jaw,

adjusting the amplitude and speed of jaw oscillations, and synchronizing the jaw movements with speech during conversations.

3. **Neck Movement Data:** This data captures the specifications for the movement of Anushka's neck module. It defines the range of motion, angles, and smooth transitions required for the robot to turn and tilt its head to establish eye contact, follow people, or express gestures.

4. **Hand Gesture Data:** This data encompasses the predefined hand gestures that Anushka can perform using its arms module. It includes gesture IDs or codes that trigger specific movements of the robot's hands, allowing it to greet, point, wave, or perform other interactive gestures.

5. **Locomotion Data:** This data governs the movement of Anushka's wheeled robot base. It includes commands for directional movement (forward, backward, left, right), duration of movement, and speed control, enabling the robot to navigate its environment and follow predefined paths.

6. **Speech and Language Data**: This data represents the linguistic inputs and outputs of Anushka's conversation module. It includes voice commands, recognized speech, natural language queries, and generated responses based on contextual analysis, semantic understanding, and the robot's knowledge base.

7. **Vision Data:** This data pertains to visual information captured and processed by Anushka's vision module. It encompasses face embeddings, hand embeddings, and object identification data used for recognizing individuals, interpreting gestures, and identifying objects in the robot's environment.

By effectively managing and utilizing these diverse data components, Anushka's communication system can process and respond to human interactions, perceive its surroundings, and engage in meaningful and dynamic exchanges.

The modules in Anushka's communication system utilize the file system as a means to transfer data and commands between each other. This file-based communication mechanism enables seamless information exchange and coordination among the different modules, facilitating the smooth operation and interaction of the robot.

4.2 Theme Setting

As stated in the section 2 of this document, Anushka, the humanoid robot, is designed with a Theme of Operations that revolves around being unbiased, caring, and intelligent, to which Anushka always caters to.
Let's explore how Anushka's modules demonstrate its commitment to these principles:

1. **Vision**:
    - Anushka should not give special attention or priority to a second person entering her vision while already engaged with someone else, maintaining an unbiased approach.
    - Regardless of familiarity, the robot should not interrupt or divert attention from the ongoing conversation to greet or acknowledge another person.

2. **Gender Neutrality**:
    - Anushka should not allocate or respond to individuals based on their gender, ensuring fair treatment and avoiding bias towards any gender.

3. **Maintaining Polite Distance:**
    - If Anushka finds herself in close proximity to another person or object of consideration she is asked to follow, she should maintain a polite distance, respecting personal space.

4. **Conversations:**
    - Anushka should nullify conversations that she is not a part of, respecting the privacy and credibility of personal interactions.
    - During conversations, Anushka should adhere to the following principles:
        - Maintain a neutral and unbiased nature of conversation based on healthy communication principles.
        - Focus on guiding the person she is talking to and providing information about KIET, the institute she represents.
        - Uphold the principle of gender neutrality, treating all individuals equally.
        - Cleverly yet politely decline inappropriate requests.

5. **Admin Mode:**
    - To ensure the confidentiality of certain administrative operations and commands, such as turning off or using Follow Me mode, Anushka should require a password along with an initiating code word to enter the admin mode.

By incorporating these principles into Anushka's design, the robot can uphold its unbiased nature, respect personal boundaries, and engage in conversations with fairness, neutrality, and professionalism.

5. COMPONENT DESIGN

The Component Design section of the Software Design Document (SDD) focuses on the detailed design of individual components that make up the Anushka humanoid robot. It provides an in-depth analysis of the internal structure, functionalities, and interactions of each component, ensuring a well-organized and modular system architecture.

In this section, we delve into the technical aspects of the robot's software and hardware components. We outline the specific design considerations, principles, and methodologies employed in developing each component, allowing for a comprehensive understanding of their roles and responsibilities within the overall system.

The Component Design section plays a crucial role in ensuring the efficient and effective implementation of the Anushka robot. It facilitates better collaboration among the development team, helps identify potential design challenges, and enables seamless integration of the individual components, ultimately contributing to the successful realization of the robot's capabilities and functionalities.

Eye Module:

```
OUTLINE:
1. Make essential Imports

2. Set eye Arduino object
3. Set initial gesture

4. Give acknowledgement to BOSS

5. Continuously check for commands on WS file
If got: transfer it to Arduino object

6. If "-1" in WS to stop, then break out of loop.
7. If error in passing to Arduino, then break out of loop.
8. Close files
```

Jaw Module:

```
OUTLINE:
1. Make essential Imports

2. Set jaw Arduino objects

3. Set initial gesture

4. Give acknowledgement to BOSS

5. Continuously check for commands on WS file
If got: transfer it to Arduino object

6. If "-1" in WS to stop, then break out of loop.
7. If error in passing to Arduino, then break out of loop.

8. Close files
```

Arms Module:

```
OUTLINE:
1. Make essential Imports

2. Set hands Arduino objects
3. Set palms Arduino object

4. Define all basic position setter functions for hands and palms
5. Define all gesture functions for hands as well as palms

6. Set initial gesture

7. Give acknowledgement to BOSS

8. Continuously check for commands on WS file
If got: Check for gesture mapping and call respective function for servo motor
manipulation

9. If "-1" in WS to stop, then break out of loop.
10. If error in passing to Arduino, then break out of loop.

11. Close files
```

```
MAJOR FUNCTIONS:
1. setValsAll() : Accepts values of servo motors for one of the both hands and
generates command to be sent to respective hand microcontroller board.

2. setAllFingers() : Accepts values of servo motor positions for one of the both hands
at a time and generates command to be sent to palm arduino.
Ensures that mirror-image phenomenon is taken care of in case of sending command to
left hand.
```

Rolls Module:

```
OUTLINE:
1. Make essential Imports

2. Set Rolls Arduino objects

3. Set initial gesture

4. Give acknowledgement to BOSS

5. Continuously check for commands on WS file

6. Check out for how much time it has to move on based on the command got
If got: transfer it to Arduino object

7. If "-1" in WS to stop, then break out of loop.
8. If error in passing to Arduino, then break out of loop.

9. Close files
```

Hearing Module:

```
OUTLINE:
1. Make essential Imports
2. Import essential APIs

3. Create 'Now.txt' file for conversation storage

4. Give acknowledgement to BOSS

5. Continuously check for commands on WS file
a. If "-1" in WS to stop, then break out of loop.
b. If "0": go to sleep and write in HearCom file to acknowledge that it has gone to
sleep
c. If "1": The hearing systems go in active mode.
Here, if s2t() helps to get user query.
Write to eye module WS to move in a random direction to show 'thinking' phase.
Translate query to English
Write the query in Now.txt file.
Check if query is any of these:
CustomCheck. SettingsCheck, CommandsCheck, BigBubble, OfflineChatbot, Chatbot

6. Close files

MAJOR FUNCTIONS:
```

```
1. queries2ChatLog(): Creates a string from the last query and result to be appended
at the end of chatlog for new query. This is essential for the AI to remember last 10
convos

2. replyBrain(): Takes a question and generates apt reply by keeping in context the
following things.
a. Knowledge base of the robot (ex- Where is EC dept., where is canteen)
b. Character file (The 'ahankara' aspect of the AI)
c. Last 5 queries (Returned from queries2Chatlog)

3. s2t(): Provides speech to text conversion using google_S2T

4. hinToEng(): Provides Hindi to English translation

5. CustomCheck(): Checks if a query is a custom check request.
Returns False if not so

6. SettingsCheck(): Checks if a query is an attempt to change the admin setting of the
robot.
Returns False if not so

7. CommandsCheck(): Checks if a given query is a direct command to the robot
Returns False if not so

8. ReplyOffline(): Generates offline response for a given query. If it is not able to
do so, return False

9. BigBubble(): Contains all the commands that can be given to the robot that add a
functionality to the robot. (ex- Application printing, weather forecasting)
```

Speech Module:

```
OUTLINE:
1. Make essential Imports
2. Initialize engine and set property to voice with speech rate= 170

3. This module will have two functions:
> speak: To only speak
> speakAndGest: To also make hand gestures while speaking corresponding to words said

4. Calculate run time for each speech output file and round it. Write this in JawWS
file
5. Wait a given time for jaw synchronization
6. If speakAndGest is called, then find the apt gesture and write to Haath file.
7. Say the sentence
8. Remoce the temp file generated for estimating time for jaw
```

Vision Module:

```
OUTLINE:
1. Make essential Imports
2. Set neck Arduino object
3. Find encodings for local people
4. Initialize camera and Detector objects

5. Give acknowledgement to BOSS

6. Continuously check for commands on WS file
If "-1" in WS to stop, then break out of loop.
```

```
Else if "0":
    Go to inactive state
else if command for mode == 1:
    Set to focus and recognize mode
    If found the admin, then change his last seen time and rewrite it in lastSeen.txt
file from Feels Module
    If a known person is recognized, then greet him according to the Robot's sense of
professionalism.
    If unknown person is recognized, then just say Hi to the person.
    Remember last 7 faces, inclusive of known and unknown people.

    Check where people are in the eye frame of the robot and randomly face them while
talking in a span of GARDAN_MOVEMENT_TIME.
    Also, if a new person enters into the frame, then look towards him/her
irrespective of GARDAN_MOVEMENT_TIME.

else if mode == 2:
    set to headshot mode
    Play a sound to acknowledge that the robot is entering headshot mode
    Try for max 3 times before a photo is discarded because it was blurred.
    If photo is success:
        Send command to hear module to go inactive.
        Ask for person to tell their name.
        Re-encode all photos and faces.
        Replay hearing module by setting it to "1" mode

else if mode == 3:
    set to Indian Sign Language recognition mode
    Find hands in the frame, at max 2
    If hands found in frame:
        Resize the frame to golden ratio
        Feed the image to classifier to check for Sign

else if mode == 4:
    set to Follow Me mode
    Call PoseDetector object to find body in the frame.
    If body found, then estimate its distance from the camera through Len’s formula
    Send adequate command to the Rolls Module based on the position of the person in
front of the robot.

7. If error in passing to Arduino, then break out of loop.

8. Close files

MAJOR FUNCTIONS:
1. findEncodings(): Finds mediapipe encodings of the face objects from images kept in
the database.
```

The Component Design section outlines the detailed specifications, interfaces, and interactions of each module, ensuring a cohesive and modular design approach. It enables clear understanding and implementation of individual components, while also facilitating seamless integration and interoperability among them. By following robust design principles and considering user requirements, the Anushka robot's components are designed to work harmoniously and efficiently, contributing to its unbiased, caring, and intelligent nature.

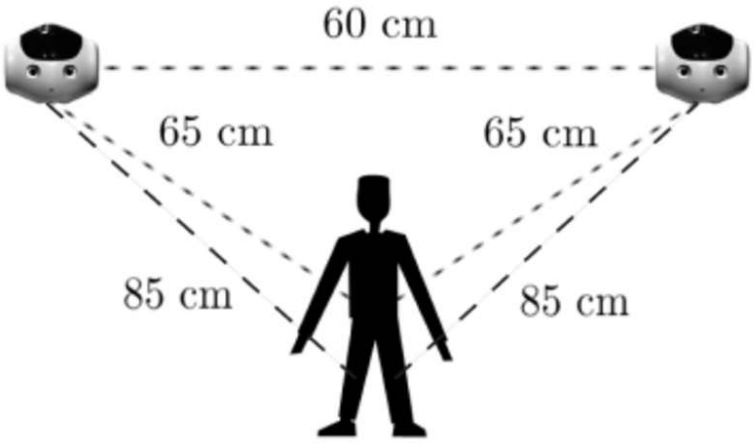
6. HUMAN INTERFACE DESIGN

The human interface design of Anushka, the humanoid robot, plays a crucial role in facilitating seamless and intuitive interactions between the robot and individuals. With a focus on user experience and usability, the design aims to create a natural and engaging interface that fosters a sense of comfort and trust. Here are the key aspects of the human interface design in Anushka:

1. **Natural Language Interaction:** Anushka leverages natural language processing and advanced speech recognition technologies to enable natural and meaningful conversations with individuals. The robot's interface is designed to understand spoken commands and respond in a conversational manner. Anushka's speech synthesis capabilities allow it to generate human-like responses, further enhancing the immersive and engaging experience for users.

2. **Visual Feedback:** To establish effective communication, Anushka's interface incorporates visual feedback mechanisms. When individuals interact with the robot, they receive visual cues, such as animated facial expressions or gestures, to indicate that their input has been acknowledged. This visual feedback helps create a sense of connection and engagement, making the interaction more dynamic and engaging.

3. **Contextual Awareness:** Anushka's interface is designed to be contextually aware, enabling the robot to adapt its responses and behaviors based on the individual's needs and the surrounding environment. The interface leverages real-time data from various sensors and modules, allowing Anushka to interpret contextual cues and tailor its interactions accordingly. This contextual awareness enhances the overall user experience and makes the interactions with Anushka feel more personalized and meaningful.



(Sense of Environment and Personal Space of Robot)

7. REQUIREMENTS MATRIX

The following requirement matrix outlines the key functional requirements for the Anushka humanoid robot project as stated in the Software Requirement Specifications Document. This matrix serves as a comprehensive reference, capturing the essential features and capabilities that the robot needs to possess. Each requirement is assigned a unique identifier, along with a description, priority level, and associated module:

Requirement ID	Requirement Description	Priority	Module
RQ001	The robot should have the ability to recognize human faces	High	Vision Module
RQ002	The robot should be able to perform natural language processing for communication	High	Conversation Module
RQ003	The robot should have the capability to move its hands in various gestures	Medium	Arms Module
RQ004	The robot should be able to move its head and neck in different directions	Medium	Neck Module
RQ005	The robot should have a mechanism for eye movements	Medium	Eye Module
RQ006	The robot should be able to generate appropriate facial expressions	Medium	Vision Module
RQ007	The robot should be able to identify and track objects	Medium	Vision Module
RQ008	The robot should be able to follow voice commands	High	Conversation Module
RQ009	The robot should maintain a safe distance from obstacles	Medium	Rolls Module
RQ010	The robot's speech synthesis should be natural and human-like	High	Conversation Module
RQ011	The robot should adhere to gender-neutral principles in its interactions	High	Conversation Module, Vision Module
RQ012	The robot should have an admin mode with password protection	High	Conversation Module
RQ013	The robot's interface should provide clear and intuitive navigation	Medium	Human Interface Design
RQ014	The robot's interface should support different accessibility options	Medium	Human Interface Design
RQ015	The robot should be able to distinguish known and unknown faces and greet known faces through their name and title.	High	Vision Module, Arms Module, Conversation Module

8. CODING STANDARDS

This section outlines the guidelines and conventions that will govern the coding practices for the development of the Anushka humanoid robot. These coding standards ensure consistency, readability, and maintainability of the codebase throughout the project.

The entire software designed and coded for the robot will follow these conventions.

8.1 Modular Conventions

Each module of the robot will have its Folder with a minimum of the following files:

1. **Combi<Module> file:** This is the actual program that carries all the code for execution of that module.
   Ex- combiJaw, combiVision etc.

2. **<Module>WS.txt:** This is the worksheet file that the program continuously keeps on checking for new commands.

3. **<Module>EF.txt:** This the error/log file that the program uses to communicate with the BOSS.

4. **Readme file:** This is the program-level documentation file that states a rudimentary algorithm along with keeping a record of various versions of the file. It should have the following content:
   - Name of the program
   - Date of combiFile creation
   - Author's name
   - Modification history
   - Synopsis (Basic Algo)
   - Functions with their input and output parameters and data types
   - Proposed global variables names
   - Proposed constants

8.2 Naming Conventions

Following are the conventions for naming various variables and tokens of the programs:

1. Name of functions to be kept in **snake_case.**
2. Name of variables to be kept in **camelCase.**
3. Name of constant to be kept in **SCREAMING_SNAKE_CASE.**

8.3 File System Conventions

Following are the conventions related to platform dependent services:

1. Folder name to be kept in **Pascal-Kabab-Case.**

Ex- My-Folder-Name
2. File name to be kept in **camelCase**.
3. File path to be stated as: **D:/Path/Of/myFile.py**

8.4 Commenting Conventions

Following are the conventions related to commenting types and their role in the program guidance:

1. **Explanation:** Regular explanatory comments are to be added using a single line comment (#), generally within the line after code, or multiline comments (“””””) generally at the starting of program

2. **Important:** Comments to denote an important statement or a command to oneself regarding the code lines near it are to be put after “#?”

3. **To-Dos:** Comments to denote to-dos that can be later altered are to be put after “#todo”.

4. **Log Alerts**: Comments to denote a monitor logging statement requirement instead of normal print statements are to be put after “#*”

5. **Error Alerts**: Comments to denote a potentially error producing statement are to be put after “#!”