SYNOPSIS

In the dynamic research environment of the ABB R&D in the Robotics and Discrete Automation division, I undertook a comprehensive internship that spanned diverse domains within the field of collaborative robotics and Natural Language Processing. As a Software Developer Intern, my primary focus was on an investigatory project to integrate voice for robotics applications, specifically integrating current functionalities of the Human Machine Interface (HMI) with a voice pipeline.

The primary motivation for embarking on the development of the Voice Assistance

Middleware is rooted in addressing the limitations of conventional human-robot interaction

methods. The motivation is threefold:

1. Operational Efficiency: Streamlining operational workflows by replacing physical buttons and text-based inputs with a more natural and efficient voice-based command pipeline.
2. User-Centric Design: Enhancing the overall user experience by introducing a hands-free and intuitive interface, reducing the learning curve for operators and enabling a broader range of users to interact with collaborative robots seamlessly.
3. Technological Advancements: Leveraging advancements in natural language processing and cloud computing to create a versatile middleware capable of intelligent interpretation and execution of voice commands.

This immersive experience honed not only my technical skills in machine learning, data analysis, and application development but also enhanced my soft skills. Effective communication was paramount as I collaborated with interdisciplinary teams and presented findings to professors. The internship demanded adaptability, problem-solving acumen, and time management, showcasing the vital role of soft skills in translating technical expertise into tangible outcomes. My journey at ABB GISPL not only expanded my technical prowess but also instilled in me a holistic understanding of the collaborative and multifaceted nature of cutting-edge research.

CHAPTER 1

PROFILE OF THE ORGANIZATION

1.1 Overview

ABB, a global technology leader, stands at the forefront of pioneering solutions in robotics, automation, electrification, and digitalization. With a rich history dating back to the late 19th century, ABB has evolved into a powerhouse of innovation, driving progress across industries and sectors worldwide. A key player in the field of robotics and discrete automation, ABB Robotics and Discrete Automation specifically focuses on delivering cutting-edge solutions that revolutionize manufacturing processes. ABB Robotics and Discrete Automation is part of the larger ABB Group, which operates across a diverse range of sectors, including utilities, transportation, and infrastructure.

ABB Robotics and Discrete Automation:

Within the ABB Group, ABB Robotics and Discrete Automation is a specialized division dedicated to transforming industries through advanced robotic and automation technologies. This division plays a pivotal role in enhancing efficiency, productivity, and flexibility for businesses involved in discrete manufacturing. The portfolio of ABB Robotics and Discrete Automation encompasses a wide range of robotic solutions, automation systems, and digital platforms. These offerings are designed to address the evolving needs of industries, ranging from automotive and electronics to food and beverage. ABB's commitment to innovation is reflected in its development of intelligent, connected, and collaborative robotic systems that redefine the possibilities of automation. The division's global presence and collaboration with industry leaders position ABB Robotics and Discrete Automation as a driving force in shaping the future of manufacturing. The focus on digitalization and smart technologies underscores ABB's dedication to providing comprehensive solutions that empower businesses to thrive in the era of Industry 4.0. As a vital component of ABB's commitment to sustainability and progress, ABB Robotics and Discrete Automation continues to push the boundaries of what is achievable in the world of automation. Through strategic partnerships, ground breaking research, and a customer-centric approach, this division exemplifies ABB's overarching mission to drive positive change through innovative technologies.

1.2 Organizational Structure

The organisational structure of ABB Robotics and Discrete Automation is as given below:

1. Chief Technology Officer (CTO):
    - At the top of the hierarchy is the CTO, responsible for setting the overall

technology strategy and direction for ABB Robotics and Discrete Automation globally.

2. Head of Robotics and Discrete Automation Worldwide:
    - Reporting directly to the CTO, the Head of Robotics and Discrete Automation oversees the division's global operations and ensures alignment with the broader organizational goals.
3. Regional Heads of R&D:
    - Positioned below the Head of Robotics and Discrete Automation, Regional Heads of Research and Development (R&D) are responsible for overseeing R&D activities in specific geographic regions. They work closely with global leadership to implement strategic initiatives and innovations tailored to regional needs.
4. Line Managers:
    - The Line Managers report to the Regional Heads of R&D and are responsible for managing specific teams or units within their respective regions. They play a crucial role in executing the R&D strategies and ensuring that projects align with the division's goals.
5. Principal Engineers:
    - Positioned below Line Managers, Principal Engineers are key technical experts within the organization. They provide leadership and guidance to other engineers, contribute to the development of innovative solutions, and play a central role in the execution of research and development projects.
6. Other Engineers:
    - Engineers in various roles work under the supervision of Principal Engineers. They contribute to project implementation, conduct research, and collaborate with cross-functional teams to bring cutting-edge technologies to fruition.
7. Interns:
    - Interns are individuals undergoing training or gaining practical experience within the ABB Robotics and Discrete Automation division. They work alongside engineers and contribute to projects while learning from experienced professionals.

This hierarchical structure ensures a clear chain of command, facilitates effective communication, and allows for efficient collaboration across different levels of the organization. It enables ABB Robotics and Discrete Automation to navigate global challenges while driving innovation in the field of robotics and automation.

1.3 Project Structure

Phases of Implementation:

Phase 1: Implementation of Voice-Based Path Teaching for GoFa Robot: In the inaugural phase, our focus lies in deploying a sophisticated voice-based system for instructing the GoFa robot. This involves streamlined command inputs to alter controller settings, activate leadthrough functionality, and create a routine module. The operational sequence initiates with precise robot positioning, capturing and storing coordinates meticulously for future use. Notably, this phase does not incorporate natural language support; instead, it relies on a curated set of candidate instructions per operation to ensure clarity and precision. Computational tasks are intelligently managed on either a Personal Computer (PC) or the Teaching Pendant Unit (TPU), strategically balancing computational load within given hardware constraints. The primary objective of Phase 1 is to establish a robust foundation for subsequent developments, paving the way for more advanced functionalities in later stages.

Phase 1.5: Introduction of Feedback Messages and Prompts: Building upon the foundation laid in Phase 1, the subsequent phase introduces feedback messages and prompts to enhance the user experience. Specifically, prompts for defining robtargets, navigating files, and adjusting jog settings will be incorporated. This refinement ensures a more interactive and user-friendly interface, addressing user needs and contributing to the overall effectiveness of the Voice Assistance Middleware.

Phase 2: Integration of Natural Language Comprehension and Multilingual Support: The second major phase focuses on incorporating Natural Language Comprehension (NLC) to streamline instruction grouping. This advancement allows operators to issue complex commands, such as creating a RAPID program to navigate between four defined robtarget points. The pursuit of Phase 2 is methodically guided by an analysis of the hardware and computational capabilities of the FlexPendant system. This includes a meticulous exploration of optimal NLP implementation strategies to ensure seamless compatibility and peak performance. Furthermore, this phase extends the functionality to embrace multiple languages and dialects, enhancing accessibility and accommodating diverse user preferences.

Timeline:

The internship took place for a period of 2.5 months between August and November 2023. It took place 5 days a week and was completely on-site.

Deliverables:

To conduct a demo on collaborative GoFa robot teaching a path using only voice commands.

1.4 Areas of Research at the Organisation

The following are the areas of research in ABB worldwide:

1. **Advanced Robotics:**

- ABB Robotics and Discrete Automation are at the forefront of developing advanced robotic systems. This includes research in areas such as collaborative robotics, precision control, and adaptive automation solutions.

2. **Automation Technologies:**

- ABB focuses on pioneering automation technologies that optimize manufacturing processes. This involves research in programmable logic controllers (PLCs), industrial robots, and intelligent automation systems for discrete manufacturing.

3. **Digitalization and Industry 4.0:**

- Embracing the principles of Industry 4.0, ABB Robotics and Discrete Automation explore digitalization strategies. This includes research in digital twins, data analytics, and connectivity solutions to create smart and efficient manufacturing ecosystems.

4. **Machine Learning and Artificial Intelligence:**

- ABB invests in research related to machine learning and artificial intelligence to enhance the capabilities of its robotic systems. This involves developing intelligent algorithms for adaptive control, predictive maintenance, and real-time decision-making.

5. **Human-Robot Collaboration:**

- ABB recognizes the importance of human-robot collaboration in modern manufacturing environments. Research efforts focus on creating robots that can work seamlessly alongside human operators, enhancing productivity and safety.

6. **Sustainable Manufacturing:**

- ABB places emphasis on sustainability in manufacturing. Research in this area involves developing energy-efficient solutions, optimizing resource utilization, and minimizing environmental impact throughout the product lifecycle.

7. I**nnovations in Discrete Automation:**

- ABB Robotics and Discrete Automation continually innovate in areas specific to discrete manufacturing. This includes research in assembly automation, quality control systems, and flexible manufacturing solutions.

1.5 Industry Collaboration and Partners

ABB's research and development thrives on a vast web of collaborators and partners, encompassing universities, startups, technology giants, and industry leaders. This collaborative spirit allows ABB to tap into diverse expertise, accelerate innovation, and bring cutting-edge technologies to market swiftly. Below is a comprehensive overview:

1. Academic Partners:

- **Strategic collaborations:** Long-term partnerships with leading universities like ETH Zurich, Stanford, and Tsinghua University focus on fundamental research in robotics, power systems, and artificial intelligence.
- **Global network:** Collaborations with universities across continents, including MIT, Imperial College London, and IIT Bombay, ensure access to a wide range of talent and research areas.

2. Industry Partners:

- **Technology leaders:** Partnerships with companies like Microsoft, IBM, and Dassault Systèmes enable integration of cutting-edge technologies like cloud computing, AI, and digital twins into ABB's solutions.
- **System integrators and consultants:** Collaborations with companies like Accenture and Capgemini help ABB deliver its solutions to customers and ensure successful implementation.
- **Startups:** ABB invests in and collaborates with startups through its corporate venture arm, ABB Technology Ventures, and its startup collaboration arm, SynerLeap. This fosters innovation and brings fresh perspectives to established markets.

3. Industry-Specific Collaborations (Robotics and Discrete Automation):

- **Research consortia:** ABB participates in research consortia like euRobotics and eu Automation to address industry-wide challenges and develop common standards.
- **Customer co-creation:** ABB actively collaborates with its customers in the robotics and discrete automation space to develop tailored solutions that meet their specific needs.
- **Open source communities:** ABB contributes to and participates in open source communities like ROS (Robot Operating System) to accelerate robotics development and adoption.

4. Additional Partnerships:

- **Science Based Targets initiative:** Collaboration with CDP, UN Global Compact, WRI, WWF, and We Mean Business Coalition to set ambitious science-based targets for greenhouse gas emissions reduction.
- **Education partnerships:** Collaborations with universities and organizations like CEMS and UNITECH International to nurture future generations of engineers and scientists.

1.6 Societal Concerns

ABB, a global leader in technology and automation, acknowledges its influence on society and grapples with various societal concerns. Here's an overview of key areas:

1. **Environmental Impact:**

- **Climate Change:** Reducing greenhouse gas emissions throughout its operations and value chain is a top priority. ABB sets ambitious carbon reduction targets and invests in renewable energy solutions.
- **Resource Efficiency:** Promoting resource efficiency and circular economy through its technologies and advocating for sustainable practices across industries.
- **Pollution Control:** Developing technologies for air and water quality improvement, while minimizing its own environmental footprint.

2. **Ethical and Labor Practices:**

- **Human Rights:** Upholding human rights within its operations and across its supply chain, addressing issues like forced labor, child labor, and discrimination.
- **Diversity and Inclusion:** Fostering a diverse and inclusive work environment, promoting equal opportunities for employees regardless of background.
- **Health and Safety:** Ensuring employee safety and well-being through strong safety programs and responsible sourcing practices.

3. **CSR Initiatives:**

- Education and Skills Development: Initiatives like "ABB Ability Tech Talks" bring technology closer to students, while scholarship programs support talented individuals.
- Environmental Sustainability: ABB sets ambitious carbon reduction targets, aiming for a 65% reduction in own greenhouse gas emissions by 2030 compared to 2019
- Resource Efficiency and Circular Economy: Initiatives like "Resource Efficiency for All" support partners in adopting sustainable solutions.
- Diversity and Inclusion: Programs like "Women in Leadership" aim to increase female representation in leadership positions

4. **Technological Impact:**

- **AI and Automation:** Addressing concerns about ethical considerations and potential job displacement arising from artificial intelligence and automation, promoting responsible development and reskilling initiatives.
- **Data Privacy and Security:** Upholding high standards of data privacy and security for customers and employees, building trust in its use of technology.
- **Digital Divide:** Bridging the digital divide by supporting programs that provide access to technology and education for disadvantaged communities.

1.7 Professional Practices

ABB, a global leader in technology and automation, fosters a unique work environment shaped by distinct professional practices. Here's a glimpse into what it might be like to work at ABB:

1. **Culture of Innovation and Collaboration:**

- **Open-mindedness and experimentation:** Encouragement for creative thinking and exploring new ideas, embracing calculated risks to drive innovation.
- **Global collaboration:** Teams across diverse locations work together seamlessly, leveraging expertise and perspectives from a wide range of cultures.
- **Flat hierarchies:** Open communication and accessibility to senior management, fostering a sense of empowerment and ownership.

2. **Focus on Sustainability and Ethics:**

- **Commitment to environmental responsibility:** Integrating sustainability principles throughout operations and developing technologies that address global challenges like climate change.
- **Ethical conduct and compliance:** Upholding high standards of business ethics and ensuring responsible sourcing practices across the supply chain.
- **Diversity and inclusion:** Building a diverse and inclusive workforce where everyone feels valued and has equal opportunities to thrive.

3. **Performance-driven and results-oriented:**

- **Emphasis on achieving ambitious goals:** Setting clear objectives and holding oneself accountable for delivering results.
- **Data-driven decision making:** Utilizing data and analytics to inform strategic choices and optimize processes.
- **Continuous improvement:** Striving for excellence through constant learning, adapting to market changes, and embracing new technologies.

4. **Work-life balance and employee well-being:**

- **Flexible work arrangements:** Offering options like remote work and flexible hours to promote work-life balance and employee well-being.
- **Investment in employee development:** Providing comprehensive training and development programs to help employees grow their skills and advance their careers.
- **Focus on health and safety:** Prioritizing employee well-being through strong safety programs and health initiatives.

CHAPTER 2

ACTIVITIES OF THE DEPARTMENT

2.1 Robotics and Discrete Automation - Overview

The Robotics and Discrete Automation division at ABB is dedicated to advancing automation solutions tailored for discrete manufacturing industries. Within this division, a multidisciplinary approach is employed, integrating robotics, automation, and digitalization technologies to enhance efficiency and flexibility in manufacturing processes. Notably, the division plays a key role in delivering solutions aligned with Industry 4.0 principles

.

2.2 Research Focus Areas

The division's research activities are centered around several key areas. This includes the development of collaborative robotics, precision control systems, and adaptive automation solutions. Research efforts also extend to the integration of artificial intelligence and machine learning in robotics for enhanced decision-making capabilities.

They develop several products like:

- Collaborative Robots – GoFa, SAY, DAY etc.
- Industrial Robots – IRB series
- Autonomous Mobile Robots – Flexley series.
- Paint Robots
- SCARA Robots

2.3 Innovation Initiatives

The division actively pursues innovation through initiatives such as the implementation of digital twin technologies, real-time monitoring solutions, and sustainable automation practices. These initiatives align with ABB's commitment to creating intelligent, connected, and energy-efficient industrial automation solutions.

CHAPTER 3

TASKS PERFORMED

3.1 Scope of Work

The work has scope in applications in robotics which involve transfer of information between a human and a robot. This allows for various scenarios of human robot collaboration.

3.2 Overview of The Work

The Voice Assistance Middleware is designed to revolutionize the human-robot interaction paradigm by introducing a more intuitive, efficient, and hands-free communication method with collaborative robots. It acts as a sophisticated extension, seamlessly integrated into the existing Teach Pendant Unit infrastructure. This innovative middleware supersedes traditional methods, eliminating the need for physical buttons and text-based inputs, offering users a more natural and convenient voice-based command pipeline. It is essentially an extension or a plug-in to the Teach Pendant Unit that will perform all the processing either on edge or using cloud services and will be seamlessly integrated into the pipeline through API calls to implement current FlexPendant functionalities like scrolling, navigating between menus and pages, configuring settings, creating robtargets and setting controller commands like enabling motors, switching between modes and enabling lead through.

The Voice Assistance Middleware is designed to revolutionize the human-robot interaction paradigm by introducing a more intuitive, efficient, and hands-free communication method with collaborative robots. It acts as a sophisticated extension, seamlessly integrated into the existing Teach Pendant Unit infrastructure. This innovative middleware supersedes traditional methods, eliminating the need for physical buttons and text-based inputs, offering users a more natural and convenient voice-based command pipeline. It is essentially an extension or a plug-in to the Teach Pendant Unit that will perform all the processing either on edge or using cloud services and will be seamlessly integrated into the pipeline through API calls to implement current FlexPendant functionalities like scrolling, navigating between menus and pages, configuring settings, creating robtargets and setting controller commands like enabling motors, switching between modes and enabling lead through.

Background and Context of the Project

In the rapidly evolving landscape of collaborative robotics, the Voice Assistance Middleware project emerges as a pivotal development poised to redefine the way humans interact with robots. As industrial processes increasingly integrate robotics for enhanced efficiency, the need for intuitive, hands-free interfaces becomes paramount. The project stems from a comprehensive understanding of this dynamic environment, where the traditional Teach

Pendant Unit, while robust, requires innovation to keep pace with evolving user expectations and operational demands.

Motivation and Context for Developing the Voice Middleware

The primary motivation for embarking on the development of the Voice Assistance Middleware is rooted in addressing the limitations of conventional human-robot interaction methods. The motivation is threefold:

1. Operational Efficiency: Streamlining operational workflows by replacing physical buttons and text-based inputs with a more natural and efficient voice-based command pipeline.
2. User-Centric Design: Enhancing the overall user experience by introducing a hands-free and intuitive interface, reducing the learning curve for operators and enabling a broader range of users to interact with collaborative robots seamlessly.
3. Technological Advancements: Leveraging advancements in natural language processing and cloud computing to create a versatile middleware capable of intelligent interpretation and execution of voice commands.

Key Objectives and Goals

The current and future features are highlighted below:

1. Seamless Integration with the Teach Pendant Unit: The middleware seamlessly integrates with the Teach Pendant Unit, extending its capabilities without disrupting the familiar user interface as it is modular and implemented as a service within the existing source code. This integration ensures that users can leverage voice commands alongside existing functionalities.
2. Enhanced Communication Efficiency: By replacing physical buttons and text-based inputs, the Voice Assistance Middleware significantly enhances communication efficiency. Users can interact with the collaborative robot using natural language commands, reducing the learning curve and operational time.
3. Flexibility in Processing: The middleware offers flexibility in processing commands, with the option to execute tasks either on the edge or leverage cloud services. This adaptability allows for optimized performance based on the specific needs and constraints of the operational environment.
4. API Integration for Existing FlexPendant Functionalities: Through API calls, the middleware seamlessly implements existing FlexPendant functionalities. This includes but is not limited to scrolling through menus, navigating between pages, configuring settings, creating robtargets, and executing controller commands like enabling motors and switching between modes.
5. Voice-Based Command Pipeline: Users can effortlessly control the collaborative robot using a voice-based command pipeline. This not only enhances operational efficiency but also reduces physical strain on operators, particularly in scenarios where manual input may be challenging or inconvenient.
6. Expanded Functionality: In addition to replicating existing FlexPendant features, the Voice Assistance Middleware introduces expanded functionalities enabled by voice

commands. This may include advanced navigation, contextual understanding, and dynamic adaptation to user preferences over time.
7. Intelligent Processing: Leveraging cutting-edge natural language processing capabilities, the middleware interprets user commands intelligently, allowing for a more nuanced and context-aware interaction with the collaborative robot.
8. Data Privacy and Authentication: With multi-user support, user authentication and encryption of data are important objectives to maintain user privacy and keep operator data safe and secure.
9. Bi-Directional Communication: To facilitate ease of use, command inputs using STT and user feedback using TTS with comprehensive and descriptive error messages, prompts and success messages to incorporate a seamless usage pipeline for an average human operator.

Scope

1. Accessible and intuitive to an average human operator, simple English commands and intuitive interfaces and prompts.
2. Adaptable with minor modifications across robots.
3. Voice assistance for as many functionalities as possible on FlexPendant.
4. Handles maximum types of errors and problems.
5. Handles ambient and noisy environments in an industrial setting.

Benefits

1. Increased Efficiency: Streamlined human-robot interaction leading to more efficient task execution.
2. Improved User Experience: Intuitive voice commands and real-time feedback contribute to a positive user experience.
3. Enhanced Security: Robust authentication mechanisms ensure secure access and control.

Primary users

The product is designed to ship with the existing FlexPendant TPU to industrial settings where collaborative robots are used on a daily basis. The intention is to make it intuitive and simple to understand for an average factory operator, but also technically sound and functionally adequate to be operated on by technician and other personnel responsible for collaborative tasks.

Stakeholders

1. Developers and System administrators – Future developers, testing and maintenance employees of the software and packages.
2. Management – Top management in companies the product is being shipped to who are the decision makers.
3. End-users – Factory-level employees and personnel operating the robot on a day-to-day basis.

Phases of Implementation

Phase 1: Implementation of Voice-Based Path Teaching for GoFa Robot: In the inaugural phase, our focus lies in deploying a sophisticated voice-based system for instructing the GoFa robot. This involves streamlined command inputs to alter controller settings, activate leadthrough functionality, and create a routine module. The operational sequence initiates with precise robot positioning, capturing and storing coordinates meticulously for future use. Notably, this phase does not incorporate natural language support; instead, it relies on a curated set of candidate instructions per operation to ensure clarity and precision. Computational tasks are intelligently managed on either a Personal Computer (PC) or the Teaching Pendant Unit (TPU), strategically balancing computational load within given hardware constraints. The primary objective of Phase 1 is to establish a robust foundation for subsequent developments, paving the way for more advanced functionalities in later stages.

Phase 1.5: Introduction of Feedback Messages and Prompts: Building upon the foundation laid in Phase 1, the subsequent phase introduces feedback messages and prompts to enhance the user experience. Specifically, prompts for defining robtargets, navigating files, and adjusting jog settings will be incorporated. This refinement ensures a more interactive and user-friendly interface, addressing user needs and contributing to the overall effectiveness of the Voice Assistance Middleware.

Phase 2: Integration of Natural Language Comprehension and Multilingual Support: The second major phase focuses on incorporating Natural Language Comprehension (NLC) to streamline instruction grouping. This advancement allows operators to issue complex commands, such as creating a RAPID program to navigate between four defined robtarget points. The pursuit of Phase 2 is methodically guided by an analysis of the hardware and computational capabilities of the FlexPendant system. This includes a meticulous exploration of optimal NLP implementation strategies to ensure seamless compatibility and peak performance. Furthermore, this phase extends the functionality to embrace multiple languages and dialects, enhancing accessibility and accommodating diverse user preferences.

3.3 Functionalities

The functionalities section outlines the core features and capabilities of the software or system. It serves to describe the main user interactions and features that users can expect from the software. This section is crucial for both technical and non-technical stakeholders, offering a comprehensive understanding of what the software can do. The idea is to introduce as many functionalities as possible, from the FlexPendant services.

Below is a list of functionalities in the current FlexPendant.

HOME:

- Startup Messages (Feedback to human operator)
    - Login/Startup Success – successful start of controller/established connection to controller.
    - Controller Details – Name, Virtual/Real, Mechanical Unit details.
- View control panel

  - Switch Mode – Manual/Manual FS/Auto.
  - Info section
  - ABB Ability section (Connected Services Summary)
  - Login/Logout/Restart Controller
- Navigate to Messages -> Clear Logs.
- Navigate to Event Log -> acknowledge errors, scroll through list, show events and errors based on category.
- Navigate to any of the options:
  - Jog
  - Program Data
  - Settings
  - I/O
  - Calibrate
  - File Explorer
- *Configure programmable buttons?*
  - *How much complex functionalities can we offer by voice.*
- Emergency Stop through voice as well.

JOG:
- Choose/Switch Mechanical unit, jog speed.
- Set Coordinate system, Tool, Work object, Load only on selecting linear jog method, else, show that it’s not selectable.
- Jog based on voice commands – Linear, Axis 1-3, Axis 4-6, Reorient.
- Select joystick movements (incremental mode) – restrict options based on mode selected. Choose angle increments.
- Navigate to Touch Jog –> Give joint number and *how much to move on that scale.
  - Do we wait for STOP JOG command from user or allow the user to specify STOPPING condition in terms of MAX angle to be reached by joint.
- Navigate to GO TO Jog -> Choose positions (robtarget/jointtarget) to navigate to.
  - If no points defined, voice command to Navigate to Program Data.
- Align TCP to appropriate frame – Choose coordinate system
- Command to reset view.
- Navigate to any other menu/functionality or event log.

SETTINGS:
- Restrict options according to Virtual/Real controller.
- Implement all features accordingly – view + edit options.
  - Network Summary – Complete Summary or specific select information like Port name/MAC Address/IP Address etc.
  - Navigate to Backup/Restore -> Choose Backup name, location and options.
  - Navigate to System Diagnostics-> Choose Filename, location and save.
  - Navigate to Time and Language -> Set it with appropriate prompts.
  - Navigate to Update
    - Feedback as ‘up to date’.
    - Upgrade software prompt from user.

  - Navigate to Smart Gripper –> set appropriate parameters.
- Navigate to any other menu/functionality or event log.

I/O:
- Navigate to I/O Networks, Devices, signals, favorite signals or configuration.
- In signals, give an option to choose amongst option of Favorite signals, digital output, digital input etc.
- Navigate to any other menu/functionality or event log.

PROGRAM DATA:
- List options for Built-in datatypes, current execution and only used datatypes or all data types.
- Let user choose from list of datatypes listed and either list details or move to the next window.
- Command to create new data element of certain type – specify name, scope, task, storage type, module, routine, dimension and then edit/copy/delete [or update position].
- Navigate to any other menu/functionality or event log.

OPERATE:
- Navigate to options of Dashboards, Advanced View, Cards, Service Routines.
  - Advanced View – Load program, Tasks and Advanced options to reset program ptr, update position, show program ptr and show motion ptr.
  - Define Dashboards – Name and Cards.
  - Define Card – Title, Label, Value, Unit.
  - Service Routines – Execute any one based on EXACT name as per list (BatteryShutdown, Commutation, LoadIdentify etc)
- Navigate to any other menu/functionality or event log.

CALIBRATE:
- Navigate to Calibration, Service Routines.
  - Calibration – Choose unit, display details.
  - Service Routines - Execute any one based on EXACT name as per list
- Navigate to any other menu/functionality or event log.

FILE EXPLORER:
- Standard file traversal based on options listed in the menu.

RAPID:
- Navigate to Program editor, Modules.
- All instructions in RAPID have to be incorporated with voice interface.
  - Add Instruction – Choose EXACT instruction from list.
  - Modify Instruction – Based on command, appropriate instruction is chosen.
  - Check Program – Read out problems/errors with program or report success.
  - Check Declarations.

  - Edit program – all options.
  - Debug Program – UI action and feedback to operator based on command (Step-in, Step-over etc)

3.4 Candidate Natural Language Instructions/Inputs

These are some examples of instructions that could be given by a human operator for each functionality.
Each screen/menu has its own set of candidate instructions that are based on functionalities offered by the existing button/text inputs.
This will help in erroneous instruction recovery/detection by utilizing already existing strategies in the FlexPendant. Additionally, it will be simpler to implement instructions where it is necessary to navigate back to another menu or have multiple cascaded instructions (eg: File Explorer). It restricts the functionality through voice to the functionality offered by the FlexPendant in a particular state (menu) which is beneficial for integration of current functionality with voice.
Each menu/screen can be visualized as a state and each candidate instruction for that state visualized as a transition to some new state. (eg: Home menu is the HOME state and, on the instruction, "Navigate to File Explorer" it moves to state FILE EXPLORER that has its own set of candidate instructions). We need to store these candidate instructions in a data structure and traverse it to see if it is an allowed instruction and if yes, we move to the next state and if no, we give appropriate error messages.
The disadvantage of this method is there will be an appreciable number of repeated instructions (eg: Navigating to HOME from FILE EXPLORER and navigating to HOME from PROGRAM DATA will be two different instructions) meaning there's more storage and space required.

HOME:

- "Start the Controller"
- "Login as administrator"
- "View Control Panel" – to open the Control Panel
  - "Switch Mode to Manual"
  - "Set Jog speed to 50%"
  - "Switch Motors ON"
  - "Give me system information"
  - "Restart Controller" – Give appropriate confirmation to user.
- "Show me the messages"
- "Open the event log"
- "Open Jog/IO/Operate/Program Data/Settings/Calibrate Menu"

JOG:

- "Switch to ROB_2 Mechanical Unit"
- "Set jogging speed to 50%"
- "Use joystick for jogging"
- "Linear jogging"
  - "Set coordinate frame as BASE"

- “Select Tool as Tool1”
- “Enable touch jog”
    - “Jog joint 1 at 20 scale until it reaches 46 degrees”
    - "Touch jog joint 3 by 20 degrees"
    - “Jog in X at -30 scale until STOP”
    - “Stop”
- “Align TCP to Work Object” – Feedback to user confirming when it’s done.
- “Jog robot till it moves to some point [x,y,z]” – Internal Loop structure to set conditions of jog.
- “Define robtarget points for GOTO jog” – Navigate to robtarget datatype in PROGRAM DATA
- "Reset the view to default" – Reset view of robot on UI.
- “Give me position information” – Feedback to user about X, Y, Z, EX, EY, EZ and coordinate frame.
- “Open IO/Operate/Program Data/Settings/Calibrate Menu”
- “View Event Log/Control Panel/Messages”

SETTINGS:
- "Open system settings"
- "Navigate to configuration options"
- “Edit controller settings”
- “Give me the network summary”
- "Check network status"
- "Create a backup by the name BACKUP1 in default location”
- "Generate diagnostic report"
- "Adjust time settings"
- "Verify software version"
- "Configure Smart Gripper"
    - “Set parameters”
    - “Calibrate gripper”
- “Open Jog/IO/Operate/Program Data/Calibrate Menu”
- “View Event Log/Control Panel/Messages”

I/O:
- “Navigate to I/O configuration”
- “Open I/O networks”/”Open I/O Devices”/”Open Signals”/”Open Favourite Signals”/”Open Configuration”
    - “Filter Favorite signals”
    - “Filter analog inputs”
    - “Filter digital outputs”
- “Open Jog/Operate/Program Data/Settings/Calibrate Menu”
- “View Event Log/Control Panel/Messages”

PROGRAM DATA:
- "List available data types"

- “Change current Execution”
  - “Set Task to ___”
  - “Set Module to ___”
  - “Set Routine to ___”
- “Display all data types”
  - “Choose robtarget”
    - “Create New Data”
    - “Edit p10”
    - “Delete p20”
  - “Choose jointtarget”
    - “Create New Data”
    - “Define coordinate frame”
  - “Display fcstring data type”
- “Open Jog/IO/Operate/Settings/Calibrate Menu”
- “View Event Log/Control Panel/Messages”

OPERATE:

- “Open Dashboard”
- “Open Advanced View”
  - “Load Program”
  - “Reset program pointer to main”
  - “Show Program Pointer”
- “Define Cards”
  - “Create New Card – Title, Label, Value, Unit”
- “Open Service Routines”
  - “Run BatteryShutdown”
  - “Run LoadIdentify”
- "Create a new dashboard"
- “Open Jog/IO/Program Data/Settings/Calibrate Menu”
- “View Event Log/Control Panel/Messages”

CALIBRATE:

- “Choose ROB_1 for calibration”
  - “Revolution Counters method”
  - “Calibrate rob_1_1, rob_1_4”
  - “Load motor calibration”
- “Open Service Routines”
  - “Run BatteryShutdown”
  - “Run LoadIdentify”
- “Open Jog/IO/Operate/Program Data/Settings Menu”
- “View Event Log/Control Panel/Messages”

FILE MENU:

- “Navigate to HOME folder”
  - “Open EDS folder”

        - “Open Vnext.modx”
        - “Edit/Copy/Remove/Rename Vnext.modx”
        - “List properties of file”
        - “Navigate up”
    - “Open TEMP folder”
        - “Navigate up”
- “Open Jog/IO/Operate/Program Data/Settings/Calibrate Menu”
- “View Event Log/Control Panel/Messages”

RAPID:
- “Open Modules”
    - “Open BASE”
    - “Open FORCECONTROL”
- “Open Program Editor”
    - “Show Declarations”
    - “Add Instruction”
        - “MoveL ToPoint ___, Speed ___, Zone ___” -> “ADD”
        - “Cancel”
        - “Wait DO Signal ___, Value ___” -> “ADD”
    - “Modify Instruction”
    - “Open Routines”
        - “Create New Routine” -> “Apply” / “Cancel”
    - “Debug program”
    - “Check Program”
- “Open Jog/IO/Operate/Program Data/Settings/Calibrate Menu”
- “View Event Log/Control Panel/Messages”

3.5 Candidate Feedback

These are potential feedback statements the middleware can give to the operator.

1. Voice activation:
    a. Success.
    b. Failure – Feedback regarding what failed from AudioCapturePermissions file.
2. Switch between auto/manual modes:
    a. Success – Acknowledge switching from manual to auto.
    b. Failure – Feedback regarding controller modes.
3. Motors on/off:
    a. Success
    b. Failure –
        i. In safety supervision mode not allowed
        ii. Safety Guard stop state
        iii. Emergency stop state
        iv. Keyless Mode Switch not permitted.
        v. Soft Safety not available.
4. Enable LeadThrough:

a. Success – Lead Through Active
b. Failure – Guard stop state, enable motors
    i. Same errors as motors on failure.
    ii. No Write Access for current user
    iii. Safety supervision needs to be set up in admin user.

5. Jogging Robot:
    a. Success – Jogging complete (in case of touch and GoTo jog)
    b. Information about active object:
        i. Mech Unit
        ii. Tool Object
        iii. Work object
    c. Feedback about which mode it is in –
        i. Joystick, Touch or GoTo.
        ii. Axis123, Axis456, Linear, Reorient
        iii. Specific Axis. Eg: Axis 4 (Action: select Axis456 and lock axes 5 and 6)
        iv. Coordinate axes being followed and position representation
        v. Speed Level
    d. Failure –
        i. Cannot jog robot in motors off state/auto mode.
        ii. Respective axis is locked, cannot permit movement (in case of locked axes)
        iii. Command is not allowed when program is running
6. Calibration:
    a. Success –
        i. Starting Calibration…
        ii. All joints calibrated
        iii. Base frame set – restart controller
        iv. Calibration points saved
        v. Calibration parameters loaded – restart controller
        vi. RAPID routine in execution
        vii. Modify/Delete/Add Module
        viii.
    b. Failure –
        i. Mechanical unit not calibrated
        ii. Report whichever joint not calibrated
        iii. Could not write values to controller
        iv. Active tool not available
        v. Controller Connection Lost
        vi. Unable to delete RAPID Data
        vii. Unable to cancel Routine
7. Program Data:
    a. Confirmation/Success:
        i. Apply/Cancel changes for robtarget.
        ii. Deletion/modification confirmation for data, routine, instruction and module.
        iii. Confirmation for creating program

   iv. Successfully modified program (cut, copy/paste etc RAPID instructions)
   v. Loading new modules will unload previous modules
   vi. Run Mode enabled
  b. Failure –
   i. Datatype doesn't exist
   ii. Repeat as not heard correctly
   iii. Variable name in current module already exists/Duplicate name/same as keyword.
   iv. Controller in manual FS mode
   v. Wrong tool/work object selected.
   vi. Functionality disabled in AUTO mode
   vii. Selected module does not contain variable data.
   viii. Arguments for variables missing/invalid
   ix. Empty placeholder, so cannot modify instruction.
   x. Points defines are too close.
   xi. Failed to update position/delete program.
   xii. Failed to load/save selected module(s).
   xiii. No robot task available/selected.
   xiv. Invalid operation
   xv. Cannot create data/program/routine/ when background task has started.

8. File Explorer:
   a. File not Found
9. Rapid:
   a. Success –
      i. PP moved to main/routine successfully
      ii. Program Pointer location
   b. Failure –
      i. Cannot add instruction due to empty placeholder
      ii. Declaration and initial value cannot be modified in Auto mode
      iii. Failed to create/copy/delete RAPID data
      iv. Failed to update position
      v. Cannot add instruction – Read/View only
      vi. Cannot edit data – Readonly
      vii. Name of module/routine/program already exists
      viii. All empty placeholders must be set to a value before expression can be written.
      ix. Robtarget cancelled
10. Miscellaneous/Generic feedback:
   a. System Failure
   b. Emergency Stop/Guard Stop
   c. Particular joint not calibrated
   d. Running Mode
   e. Manual full speed state

3.6 Candidate Prompts

1. Calibration:
    a. Select method to define Base frame
    b. Cancel Service Routine
    c. Label to choose axes to calibrate
2. Jogging:
    a. Choose frame to align
    b. Choose Payload for robot
    c. Press and hold the button for GoTo jog
    d. Invalid Jog Parameters
3. ProgramData:
    a. Cancel/accept changes
    b. Cancel current operation
    c. Change datatype of selected data
    d. Enter new name
    e. No routines in current module, create routine
    f. Create a New Module
    g. Replace current loaded program
    h. Press and hold enabling device
    i. Restore Tool Define/Workobject define
    j. Use keyboard to modify text
4. RAPID/Code:
    a. Fill parameters in for ADD INSTRUCTION (MoveJ -> Point, speed, tool, optional arguments)
    b. Create New Module (no module current loaded)
    c. Create a new Routine
    d. Creating new programs will delete loaded ones, wish to proceed?
    e. No module exists for creating new data, create a module?
    f. Program pointer will be lost, proceed?

3.7 Future Features

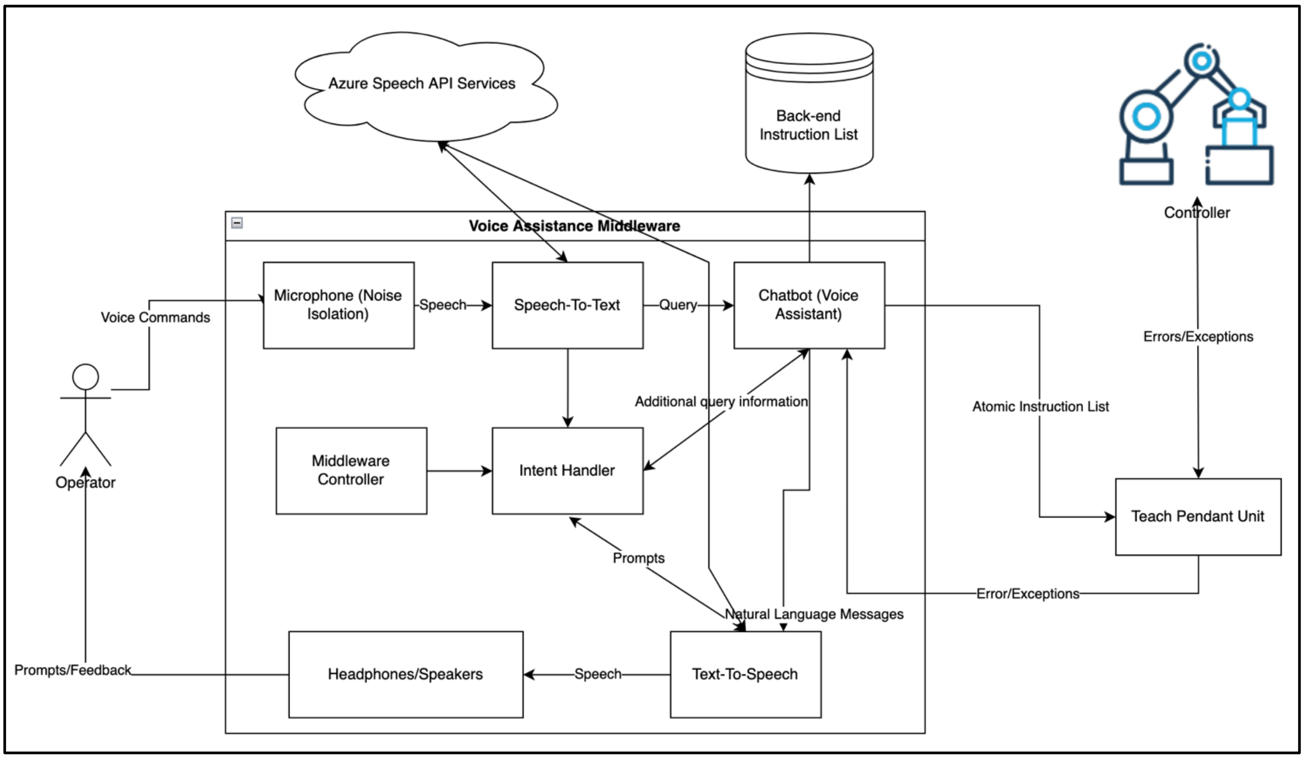
1. User authentication – Support multiple users and allow for the seamless switching of user profiles to personalize the interaction. Secure user authentication to prevent leak of passwords and user data.
2. Task Scheduling - Enable users to schedule tasks for the collaborative robot at specific times, enhancing automation capabilities.
3. Real-time feedback and prompts - Provide real-time feedback to the user during task execution to keep them informed of the robot's progress.
4. Error Handling - Robust error handling to manage situations where the robot cannot fulfill a command and provide informative error messages and visual error messages.
5. Machine Learning integration – Integration of chatbots to create a virtual voice assistant to process natural language instructions and adapt to users' patterns over time and to improve accuracy.
6. Localized language support – Support multiple regional languages with a chatbot backend and increase accessibility.
7. Logging and Debugging – Voice-based data logging and debugging to improve testing performance and increase convenience to users.

8. Cloud services integration - Leverage Azure services for natural language processing, computation, and potentially other cloud-based functionalities.

3.8 Architecture and Design

The architecture and design section provide insight into the high-level structure and design principles of the software system. It explains the relationships between different components, architectural patterns, and design principles employed in the development process. This section is essential for understanding the system's overall structure.

Proposed Architecture Diagram



Below is a brief description of each of the modules that have been mentioned in the diagram above.

1. Microphone (Noise Isolation):

- **Description:**
    - The microphone captures voice commands from the operator, serving as the primary input device for the Voice Assistance Middleware.
    - The microphone can either have in-built noise isolation (JABRA Headsets) or a separate engine for noise isolation/cancellation.
- **Key Features:**
    - Includes noise isolation capabilities to enhance the accuracy of speech recognition in varied environmental conditions.
    - Reduces ambient noise and isolates operator voice commands in noisy environments.

2. Speech to Text (STT):

- **Description:**

- Converts the captured speech from the microphone into text, making it interpretable by the Voice Assistance Middleware.
- **Key Features:**
  - Utilizes speech recognition algorithms to accurately transcribe spoken words into written text, from Azure cloud services in order to offboard processing to the cloud and not on the device as it has limited processing power.

3. Text to Speech (TTS):

- **Description:**
  - Converts text-based responses generated by the Voice Assistance Middleware into spoken words, providing audible feedback to the operator.
- **Key Features:**
  - Employs natural-sounding synthesis for a more human-like voice output.
  - Also uses Azure cloud services to convert natural language instructions to speech based on regional language for easy interpretability of operator.

4. Cloud Services:

- **Description:**
  - Represents external cloud-based services that may be leveraged for natural language processing, voice recognition, or other advanced functionalities.
- **Key Features:**
  - Integration with cloud services enhances the Voice Assistance Middleware's capabilities, allowing for scalable and sophisticated processing.

5. Chatbot (Voice Assistant):

- **Description:**
  - Interprets user queries and commands, providing relevant responses and instructions.
- **Key Features:**
  - Incorporates natural language processing to understand and respond to user inputs in a conversational manner.
  - Communicates with a data-store either offline or online to map natural language inputs to backend instruction set.
  - Allows for compound natural language instructions to be given that will convert to set of atomic instructions in sequence to be sent to TPU.

6. Middleware Controller:

- **Description:**
  - Orchestrates the flow of information between various components, ensuring seamless communication and coordination within the Voice Assistance Middleware.
- **Key Features:**
  - Manages the interaction between the user, microphone, STT, chatbot, TTS, and other modules.
  - Manages the sequential processing of the instruction list compiled by chatbot to send to HMI as atomic instructions after completion of each instruction.

7. Intent Handler:

- **Description:**

  - Evaluates user queries to determine the intent and coordinates with the chatbot for appropriate responses. Handles requests for re-prompts with additional information.
- **Key Features:**
  - Enhances user experience by dynamically adapting to user queries and providing contextual responses.
  - Handles authentication and multiple entries ensuring operator doesn't have to know exact format for various instructions and can give basic instructions and be prompted for more instructions.

8. Speakers:

- **Description:**
  - Outputs the synthesized speech generated by the Text to Speech (TTS) module, delivering voice feedback to the operator.
- **Key Features:**
  - Provides clear and audible responses, contributing to a user-friendly and effective interaction.

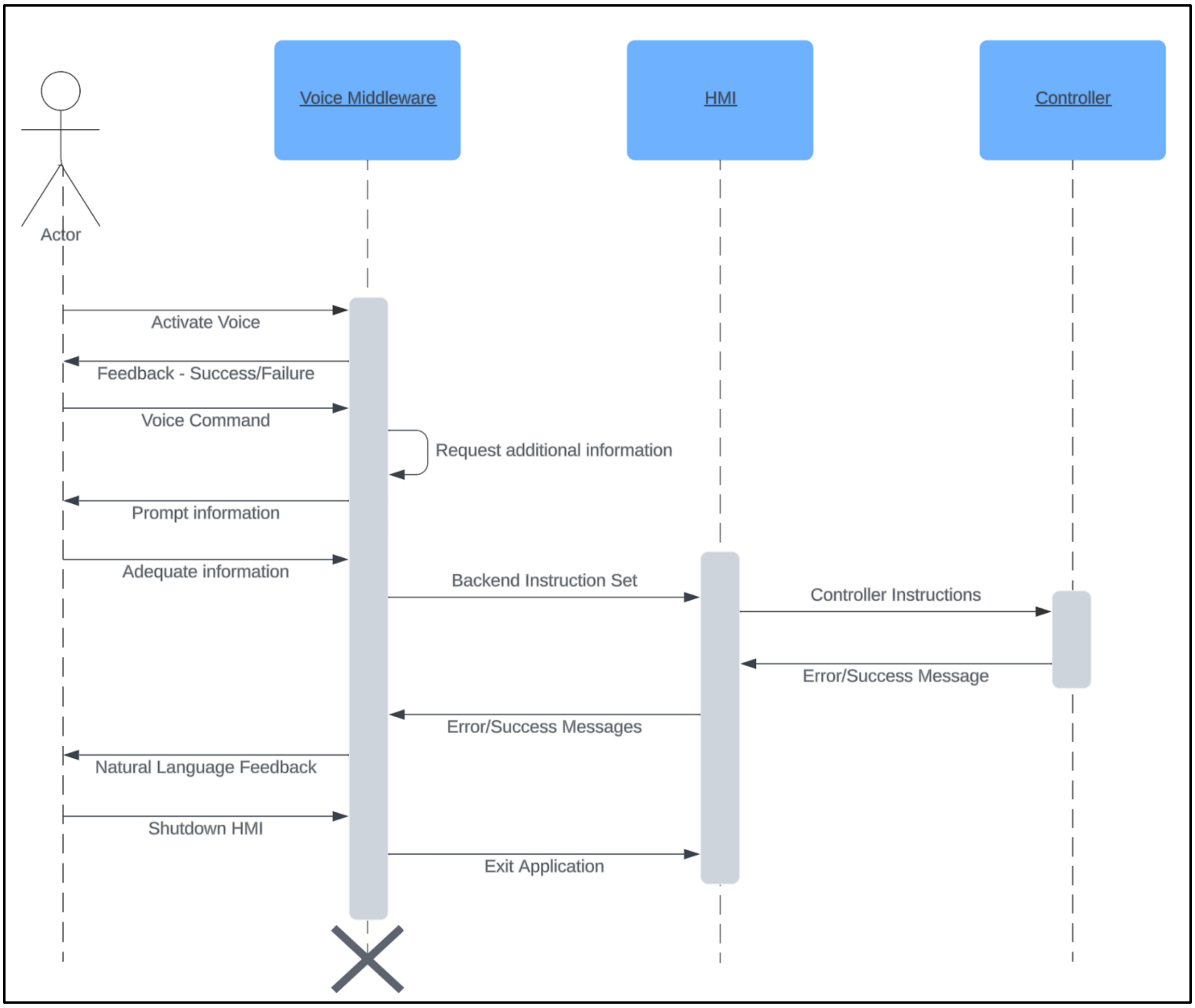
Relationships Between Key System Elements:

- **Microphone to STT:**
  - Microphone captures speech, and STT converts it into text for further processing.
- **STT to Chatbot:**
  - Text-based queries from STT are sent to the chatbot for interpretation.
- **Chatbot to HMI:**
  - Responses and instruction sequences from the chatbot are displayed on the HMI for the operator.
- **HMI to Chatbot (Error Handling):**
  - Error messages and user inputs from the HMI are interpreted by the chatbot for appropriate responses.
- **Chatbot to TTS:**
  - Natural language responses from the chatbot are converted into synthesized speech by the TTS module.
- **Intent Handler to Chatbot and TTS:**
  - Coordinates with the chatbot for responses and communicates with TTS for additional voice feedback based on user intent.
- **Middleware Controller:**
  - Orchestrates the overall flow of information between the user, microphone, STT, chatbot, TTS, HMI, and other modules.

3.9 Behavioural Characteristics

The behavioural characteristics section focuses on the dynamic aspects of the system, illustrating its behaviour in various scenarios. It includes visual representations such as sequence diagrams and activity diagrams, accompanied by detailed descriptions and narratives that explain the system's responses to different user interactions.

Sequence Diagram



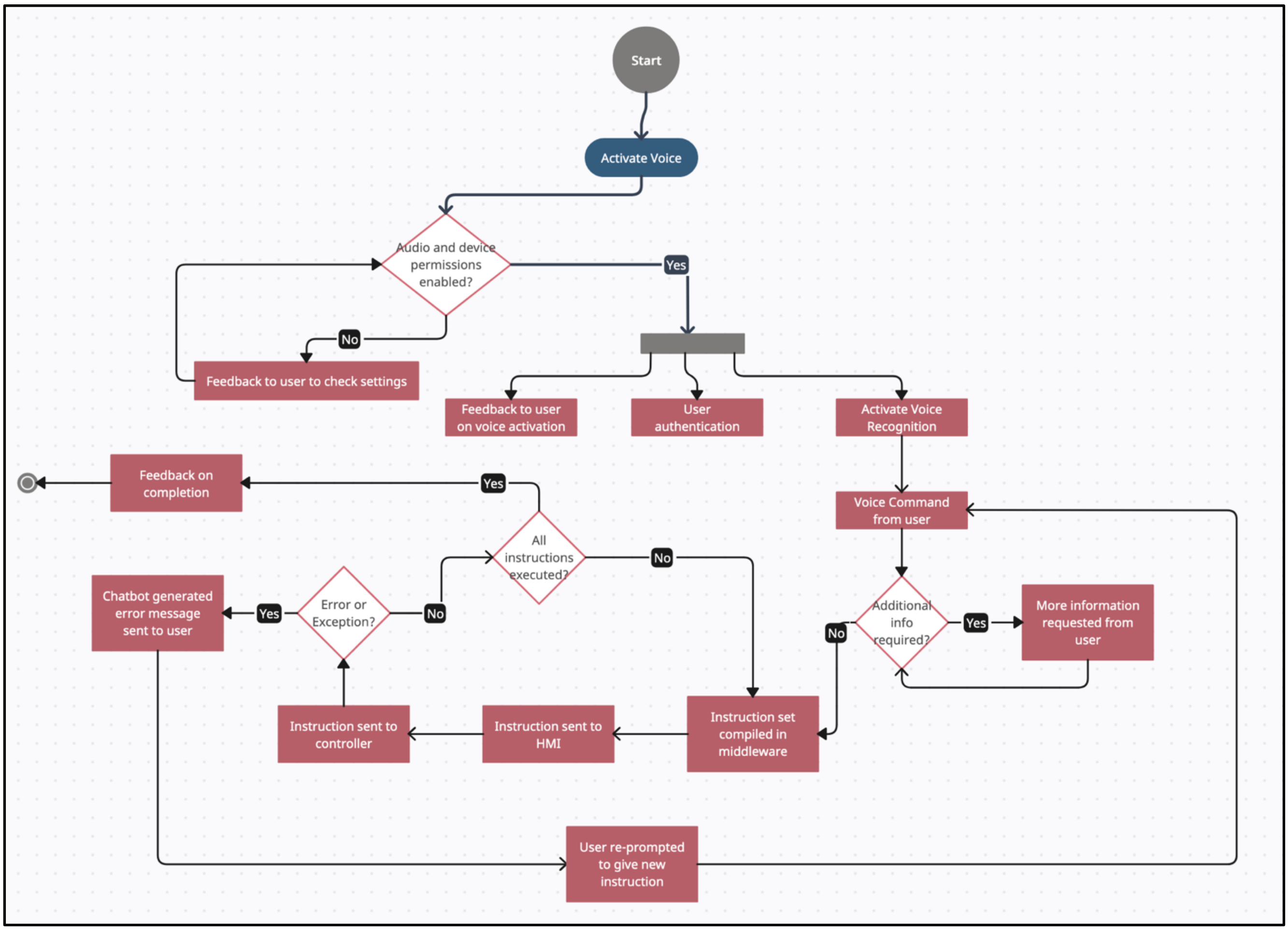
Below is the description of the sequence diagram and its key elements.

1. **Actor Request - System Activation:**
   - The sequence initiates when the operator issues a verbal command, "Activate voice," triggering the voice middleware activation process.
   - The activated voice middleware establishes communication with the HMI, signalling its readiness to receive commands.
2. **Command Execution - Voice Middleware to HMI to Controller:**
   - Upon receiving voice commands from the operator, the voice middleware interprets the instructions and forwards them to the HMI for execution.
   - The HMI relays these instructions to the controller for further processing and execution within the system.
3. **Error/Failure Handling - Controller Response:**
   - In case of successful command execution, the controller processes the instructions and provides feedback to the HMI for relay to the operator as natural language prompts.

- Conversely, if errors or failures occur during command execution, the controller generates corresponding error messages.

4. **Feedback Loop - Natural Language Feedback to Operator:**
   - The natural language feedback, generated by the controller, is relayed back through the HMI to the operator, informing them of the command's success or failure.
5. **System Deactivation - Operator Request:**
   - The sequence culminates when the operator issues the command, "Deactivate the system," initiating the shutdown process.
   - The command traverses through the system components, leading to the deactivation of the voice middleware and subsequent components.

Activity Diagram



This activity diagram represents the workflow of the voice assistant middleware system, encompassing the activation, command execution, error handling, and deactivation processes.

1. **Start - System Activation:**
   - The process begins with the activation of the voice assistant system when the operator initiates the "Activate voice" command.

- The activation process involves initializing the voice middleware, establishing communication with the HMI, and waiting for user commands.

2. **Receive and Process Voice Commands:**
    - The system continuously waits for voice commands from the operator.
    - Upon receiving a command, the voice middleware processes and interprets it before passing the instructions to the HMI for execution.
3. **Execute Commands - HMI to Controller:**
    - The HMI relays the interpreted commands to the controller for execution within the system.
    - The controller processes the commands, interacting with relevant components to carry out the requested actions.
4. **Handle Errors - Error/Failure Path:**
    - If errors or failures occur during command execution, the system follows an error path.
    - The controller generates corresponding error messages, and the system communicates these back to the operator through the HMI.
5. **Provide Feedback - Natural Language Feedback:**
    - In case of successful command execution, the controller generates feedback messages.
    - The HMI relays these messages back to the operator in natural language, providing prompts or information about the executed command.
6. **End - System Deactivation:**
    - The operator can initiate the system's deactivation by issuing the "Deactivate the system" command.

The deactivation process involves shutting down the voice middleware and related components, concluding the system operation.

3.10 Functional Requirements

Functional requirements outline specific features, capabilities, and behaviors that the software must exhibit to meet user needs and business objectives. They serve as a detailed specification of the system's functionalities, guiding the development team in building a product that aligns with user expectations.

Below are the functional requirements identified:

1. **Voice Command Recognition:**
    - *Description:* The system must accurately recognize and interpret voice commands from the operator.
    - *Acceptance Criteria:* Achieve a recognition accuracy of at least 95% for a predefined set of commands.
2. **Natural Language Processing (NLP):**
    - *Description:* Implement NLP capabilities to allow the system to understand and respond to natural language queries.
    - *Acceptance Criteria:* Enable the system to handle diverse user inputs and provide meaningful responses.

3. **Cloud Service Integration:**
    - *Description:* Integrate with cloud services, such as Azure, for advanced processing and data storage.
    - *Acceptance Criteria:* Successfully connect and communicate with specified cloud services through secure APIs.
4. **Error Handling and Feedback:**
    - *Description:* Provide clear and informative error messages to the operator in case of unrecognized commands or system issues.
    - *Acceptance Criteria:* Error messages should be contextual and guide the user on how to correct the issue.
5. **Multilingual Support:**
    - *Description:* Support interaction in multiple languages and dialects.
    - *Acceptance Criteria:* Allow the operator to seamlessly switch between supported languages during interaction.
6. **Intent Handling:**
    - *Description:* Implement an intent handler to manage additional information requests and responses with the Chatbot and TTS module.
    - *Acceptance Criteria:* Successfully coordinate and communicate between the Chatbot and TTS for enhanced user interaction.
7. **User Authentication:**
    - *Description:* Authenticate the user to ensure secure access and prevent unauthorized use.
    - *Acceptance Criteria:* Implement secure authentication mechanisms, such as user credentials or biometric authentication.

3.11 Non-Functional Requirements

Non-functional requirements focus on the qualities and characteristics that define the overall behavior and performance of the software. Unlike functional requirements, which specify what the system should do, non-functional requirements define how well the system should perform those functions. They address aspects such as performance, security, usability, and reliability.

Performance, Scalability, and Reliability Requirements:

1. **Performance:**
    - The system should respond to voice commands within 1.5-2 seconds.
2. **Scalability:**
    - The system should handle a minimum of 10 concurrent users and robots.
3. **Reliability/Availability:**
    - The system should have an uptime of at least 90%.

Security and Privacy Considerations:

1. **Data Encryption:**
    - All communication between the Voice Middleware and cloud services should be encrypted using secure protocols (e.g., HTTPS).

2. **User Privacy:**
    - The system should adhere to privacy regulations and ensure that user voice data is handled securely.
3. **Access Control:**
    - Implement role-based access control (admin and default user) to restrict access to sensitive functionalities.

Compatibility Requirements:

1. **Cloud Service Compatibility:**
    - Ensure compatibility with specified versions of cloud services, including Azure.
2. **Device Compatibility:**

   Support a range of devices, including different microphones and speakers commonly used in industrial settings.

3.12 Risk Analysis

The risk analysis section identifies potential risks that may impact the success of the project and proposes strategies to mitigate or manage them. This section provides an analysis of potential risks, such as technical challenges or external dependencies, offering insights into how these risks might be addressed to ensure project success.

Identification of Potential Risks

1. **Speech Recognition Accuracy:**
    - *Risk:* The system may struggle with accurate speech recognition, leading to misinterpretation of user commands.
    - *Likelihood:* High
    - *Impact:* Medium
    - *Mitigation:*
        - Conduct thorough testing with diverse user voices and accents during the development phase.
        - Implement continuous improvement mechanisms for the speech recognition algorithm based on real-world usage data.
2. **Cloud Service Outages:**
    - *Risk:* Dependence on external cloud services may expose the system to downtime during service outages.
    - *Likelihood:* Medium
    - *Impact:* High
    - *Mitigation:*
        - Implement a failover mechanism to switch to alternative cloud services in case of a primary service outage.
        - Regularly monitor the status of cloud services and have contingency plans in place.
3. **Security Breaches:**
    - *Risk:* Unauthorized access to user data or system functionalities.
    - *Likelihood:* Medium

- *Impact:* High
- *Mitigation:*
  - Implement robust encryption for data in transit and at rest.
  - Conduct regular security audits and penetration testing to identify and address vulnerabilities.

4. **User Authentication Issues:**
   - *Risk:* Challenges in implementing effective user authentication mechanisms.
   - *Likelihood:* Medium
   - *Impact:* Medium
   - *Mitigation:*
     - Utilize industry-standard authentication protocols.
     - Provide clear user guidelines for secure authentication practices.
5. **Compatibility Challenges:**
   - *Risk:* Issues with compatibility across different devices and browsers.
   - *Likelihood:* Medium
   - *Impact:* Low to Medium
   - *Mitigation:*
     - Test the system on a variety of devices and browsers during development.
     - Maintain a compatibility matrix and regularly update it based on user feedback and changing technologies.

Mitigation Strategies

1. **Iterative Development and Testing:**
   - Regularly conduct iterative testing, incorporating user feedback to enhance the accuracy of speech recognition.
2. **Diversification of Cloud Services:**
   - Utilize multiple cloud service providers or have contingency plans to switch services in case of an outage.
3. **Comprehensive Security Measures:**
   - Regularly update and patch system components to address security vulnerabilities promptly.
   - Educate users on security best practices to minimize the risk of unauthorized access.
4. **User Authentication Best Practices:**
   - Implement multi-factor authentication to enhance user authentication security.
   - Regularly review and update authentication mechanisms based on evolving security standards.
5. **Continuous Compatibility Testing:**
   - Maintain an ongoing compatibility testing process to address issues promptly as new devices and browsers emerge.
   - Establish a feedback mechanism for users to report compatibility issues.
6. **Risk Monitoring and Reporting:**

- Implement a robust risk monitoring system to detect and report potential risks in real-time.
- Establish a clear communication plan for stakeholders in case of identified risks.

7. **Regular Training and Awareness Programs:**
    - Conduct regular training sessions for system users to raise awareness about potential risks and security best practices.

CHAPTER 4

REFLECTIONS

Having worked on the project for about two months, from the starting phase of learning, then building and fixing errors in the design and code, the Internship gave me a lot of exposure to the process of development. The people I got to work with helped me in every step of the development process, and gave me a good understanding of what it means to be working together as a team to solve problems.

4.1 Technical Knowledge Acquired

Below are the technical skills I acquired during the course of my internship at ABB Robotics and Discrete Automation:

1. Programming Languages and Frameworks:

- Developed the Voice-Assistant Middleware using Microsoft C# programming language within the .NET programming framework.
- Worked with the Windows UWP (Universal Windows Platform) app architecture.

2. Integrated Development Environment (IDE):

- Utilized Microsoft Visual Studio as the primary development platform.
- Leveraged Visual Studio's debugging tools and various features for efficient code development.

3. Collaborative Robot and Software:

- Acquired hands-on experience with ABB's collaborative robot, GoFa.
- Worked extensively with ABB's proprietary software, including RAPID and Robot Studio.

4. Robot Control Interfaces:

- Implemented functionalities using the FlexPendant interface for programming and controlling the GoFa robot.
- Executed tasks directly on the robot using the FlexPendant as the main source of implementation.

5. Speech Recognition Toolkit:

- Implemented voice control using the Microsoft Speech Recognition toolkit.
- Processed speech commands based on predefined grammars within the Voice-Assistant Middleware.

6. ABB Robotics-Specific Code:

- Worked with ABB proprietary code, including RAPID programming language and Robotstudio environment.

- Interfaced the Voice-Assistant Middleware with ABB-specific technologies.

7. Robot Teaching through Voice Commands:

- Applied practical knowledge by teaching the GoFa robot paths through voice commands.
- Developed an understanding of natural language-based human-robot interaction.

8. Documentation:

- Created a comprehensive vision document documenting the development process.
- Captured the implementation details, challenges, and solutions for future reference.

9. Human-Robot Interaction (HRI):

- Gained insights into designing and implementing solutions for effective Human-Robot Interaction (HRI).
- Explored the integration of voice commands to enhance the user experience with collaborative robots.

10. Path Planning and Implementation:

- Executed hands-on work involving the GoFa robot's path planning through voice commands.
- Translated high-level voice instructions into actionable paths for the robot.

11. Natural Language Processing (NLP):

- Contributed to the development of a vision document as a stepping stone for natural language-based interaction.
- Explored the potential of integrating NLP concepts for future advancements.

4.2 Soft Skills Acquired

Embarking on my internship journey at ABB GISPL, I not only expanded my technical prowess but also honed an array of indispensable soft skills crucial for personal and professional growth. Engaging in virtual networking and communication, I navigated the intricacies of remote collaboration, ensuring seamless interactions with team members and stakeholders. Juggling college commitments alongside my internship, I cultivated a harmonious college-work life balance, a testament to effective time management. Embracing teamwork and collaboration, I collaborated across interdisciplinary teams, understanding the significance of diverse perspectives in achieving comprehensive project outcomes. The experience instilled in me a proactive approach, as evidenced by my readiness to take initiative in various facets of project development. Exhibiting professionalism in every interaction, I learned the nuances of effective communication and workplace etiquette. Furthermore, the dynamic nature of the internship fostered adaptability, teaching me to navigate challenges with resilience. These soft skills, as integral components of personal and professional development, are not only essential for success in a collaborative work environment but also serve as the bedrock for continuous learning and growth.