UPANZI
DIGITAL PUBLIC INFRASTRUCTURE
NETWORK

Pepper Robot Tour for Upanzi's Digital Experience Center

Work Plan

Version 1.6
June 02, 2026

Start date of project: **01/07/2025**

Duration: **12 months**

Partner organisations: **Carnegie Mellon University Africa**

Project funded by the CyLab-Africa/Upanzi Network
Digital Experience Center

Executive Summary

This document presents a structured tweleve-month work plan to develop a culturally informed human–robot interaction system for the Digital Experience Center (DEC) at CMU-Africa's Upanzi lab. The project will:

1. **Adapt** the existing CSSR4Africa (`https://cssr4africa.github.io/`) social-robotics framework to meet the specific needs of Upanzi's DEC.
2. **Survey** and evaluate current DEC projects, exhibits, and visitor-engagement features.
3. **Design** and implement scenario-driven control scripts enabling Pepper robots to:
    - Greet visitors upon arrival.
    - Guide tours through key exhibits.
    - Highlight and explain featured DEC offerings.
    - Incorporate a Large Language Model (LLM) to enable interactive Q&A with visitors.
4. **Align** all robot behaviors and dialogue with Rwandan social norms and expectations to ensure interactions feel natural, respectful, and culturally appropriate.

Contents

Chapter 1

Project Description

1.1 Motivation

Figure 1.1: The Digital Experience Center (DEC) at CMU-Africa's Upanzi lab, featuring interactive role-play booths that simulate the lifecycle of Digital Public Infrastructure.

The Pepper Robot Tour for the Digital Experience Center is a spin-off project of the Culturally Sensitive Social Robotics for Africa Project (CSSR4Africa) [1], driven to embody the Upanzi Digital Experience Center in digital public infrastructure through an end-to-end automated tour. In addition, to push frontiers of cultural aware human-robot interaction in real-world settings. The goal is to replace repetitive, human-led walkthroughs with an autonomous Pepper robot that is capable of role-playing and interacting with Upanzi Guests. This project is motivated by two goals. (i) Authentic Digital Immersion: The Upanzi Republic story is a fictitious nation designed to educate visitors on the journey of digital identity creation and use cases

[1] `https://cssr4africa.github.io`

such as government service enrollment and subsidy disbursement using a role-play approach. Its interactive role-play booths help visitors viscerally experience the life cycle of Digital Public Infrastructure (DPI). Yet, when humans repeatedly enact these roles, the very digital narrative we're promoting can feel less convincing. By replacing the facilitators with autonomous Pepper robots, every step of the tour, from biometric capture at the MOSIP booth through a real-time MIFOS transaction to subsidy validation in UPMS (Upanzi Programs Management System), it becomes seamless. (ii) Building on the CSSR4Africa's work, DEC's Pepper robots will speak, gesture, and engage in local dialects that resonate with diverse audiences. In running this tour, visitors' engagement at each booth will be used to refine cross-cultural robot behavior, adaptive dialogue management.

1.2 Technical Scope

The Robotics for Digital Experience Center (DEC) project builds on the software and packages developed under CSSR4Africa, introducing targeted adaptations to meet the unique requirements of an autonomous, Pepper-driven tour environment. The following section summarizes how each major work package (WP) from CSSR4Africa is adapted for the DEC context:

Work Package Adaptations

Work Package	Inherited from CSSR4Africa	DEC Adaptation
WP1: DEC Tour Guide: Initial CSSR4Africa ROS1 version	CSSR4Africa robotic lab tour script.	Develop a tour script for the Digital Experience Center using the current CSSR4Africa software.
WP2: Interaction Scenario Specification	Use case scenario definition templates and dialogue flows.	Map DEC booths into scenario definitions (Task 2.1). Specify Pepper's behavior and visitor state at each station (Tasks 2.2–2.4) using the tour scripts.
WP3: Systems Engineering	Core system architecture.	Extend the architecture by integrating a conversation manager to drive natural interaction between visitors and the Pepper robot.
WP4: Robot Sensing	Sensor drivers and perception nodes for cameras, LiDAR, and microphones.	*(1) ROS1 → ROS2 Migration:* Port existing perception and sensor driver packages to ROS2 Humble, updating message definitions, launch files, and deprecated APIs. *(2) SLAM-based Localization:* Replace ArUco marker-based localization with a SLAM solution (e.g., RTAB-Map) on ROS2 for real-time map building and fiducial-free localization within the DEC.
WP5: Robot Behaviors	Actuator control, basic gestures, and speech pipelines.	*(1) ROS1 → ROS2 Actuator Migration:* Port motion primitives and hardware interfaces to the ROS2 control framework. *(2) Gesture Animation:* Add culturally appropriate gestures and expressions with coordinated sound/motion timing (Task 5.4). *(3) Behavior Controller:* Implement a behavior tree or state machine to sequence tour role-play states, integrating the cultural ontology (Task 5.2). *(4) Navigation with Nav2:* Deploy the Nav2 stack using the SLAM-generated DEC map for global planning and obstacle avoidance (Task 5.3).

Work Package	Inherited from CSSR4Africa	DEC Adaptation
WP6: Use Case Evaluation	Evaluation protocols and data collection tools.	Deploy DEC scenarios with Pepper (Task 6.1) and conduct visitor studies to measure engagement, comprehension, and response quality (Task 6.2).
WP7: Dissemination and Impact	Open-source release processes and dissemination templates.	Create a dedicated DEC GitHub repository (Task 7.3), update the project online presence to feature the autonomous tour (Task 7.1), and organize workshops and presentations for the Upanzi network team and the broader CMU-Africa community (Task 7.2).

By structuring the DEC's development around these adapted work packages, we leverage the proven CSSR4Africa foundation while making targeted modifications to realize a fully digital, Pepper driven tour experience.

Chapter 2

Work Plan

2.1 Approach and Activities

We adopt both a user-centric perspective and an agile and iterative approach in this project. This is reflected in the work plan; see the Pert chart in Figure 2.1 and the list of work packages in Table 2.1.

Table 2.1: List of work packages

WP No.	Work Package Title Partner	Person Months	Start Month	End Month
1	DEC Tour Guide: Initial CSSR4Africa ROS1 version	2	1	1
2	Interaction Scenario Specification	5	1	12
3	System Architecture & Systems Engineering	6	1	12
4	Robot Sensing	6	1	12
5	Robot Behaviors	4	1	12
6	Use Case Demonstration and Evaluation	3	10	12
7	Dissemination and Impact	6	1	12

WP1 is user-driven and focuses on identifying the cultural and social norms, i.e., cultural knowledge, that define respectful and engaging interaction in Rwanda. This has been achieved through the use of ethnographic user studies carried out in the CSSR4Africa project. The data generated define the development of Rwanda-centric modes of human-robot interaction and design patterns for courteous and respectful sociality. Culture knowledge and design patterns are used to specify the robot behaviors in WP5, which are used in the demonstration and evaluation use case in WP6. Therefore, WP1 here involves reviewing the cultural knowledge base of CSSR4Africa and deciding what is relevant in the context of providing a tour in upanzi DEC. The robot sensing functionality required for interaction is developed in WP4. The integration of all functionality into a coherent system architecture is carried out in WP3, while the specification of the interaction scenarios is carried out in WP2. The interpretation of these scenarios is affected by the robot behaviors subsystem developed in WP5. Monitoring research progress, meanwhile, is also done in WP6 through user studies that test and validate the targeted use case functionality at the end of the fourth month of the project, taking appropriate action after the fourth month to adjust and augment each element in WP1 - WP5 in order to improve the performance in the use cases in the subsequent phase. The timeline also highlights this iterative development; see the Gantt chart in Figure 2 in Section 2.4, where detailed descriptions of the work packages are also provided. All the work packages in this project will build on the work already done in the CSSR4Africa project.

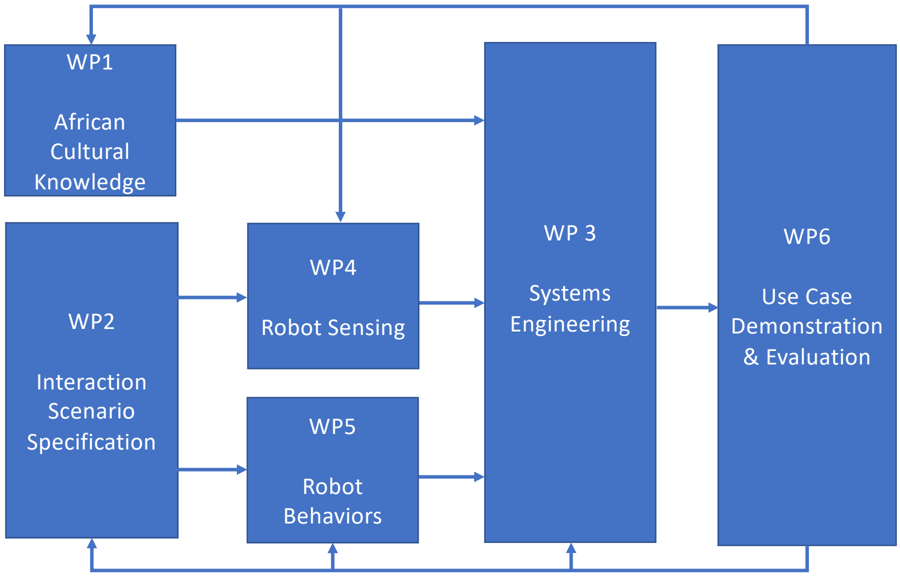

Figure 2.1: PERT chart showing the dependencies between the technical work packages; note the iterative development cycle.

From a software engineering perspective, this project will adopt current best practice in robot software development based on component-based robotic engineering (Brugali & Scandurra, 2009; Brugali & Shakhimardanov, 2010) and adapted from established component-based software engineering of component-based software engineering (Heineman & Council, 2001; Szyperski, 2002). With a focus on effective integration, the CSSR4Africa system will adhere to the best practice of making components composable (the property that makes it easily integrated into a larger system) and systems compositional (the property of exhibiting predictable performance and behavior if the performance and behavior of the components are known).

We will use ROS, a globally-used implementation of component-based software engineering (CBSE). Furthermore, we will adopt an integration-focused approach to the development of the system architecture (Vernon et al., 2015) based on CBSE, in general, and the component-port-connector model (i.e., the publish and subscribe model), in particular. In essence, then, we propose an adaptive, compositional agent-based message-passing software architecture to bridge WP4 & WP5 functionality and WP6 use-case behaviors.

All software will adhere to a project-specific set of software specification, design, coding, and documentation standards. These standards will also be applied to every module or subsystem that comprises each of the two primary subsystems (robot sensing and robot behaviors). In this way, the complete system will be based on a modular decomposition with the same interface protocol and software engineering standards applying at every level of decomposition. This approach facilitates transparent configuration and incremental integration & test of the complete system, from component, to subsystem, to system as a whole.

This approach ensures that those responsible for these subsystems have sufficient freedom to choose the design that suits the subsystem needs best while at the same time requiring them to adhere to project-wide standards of software engineering and quality assurance.

2.2 Expected Outcomes

2.3 General Project Information

Facilities

We will use the Pepper social robot from Aldebaran, a part of the United Robotics Group, to develop an automated robot guided tour for Upanzi's Digital Experience Center.

Measures of Success and Assessment

To evaluate the tour, we will follow the CSSR4Africa's project frame by using the Robot Social Attribute Scale (RoSAS). RoSAS captures three key dimensions of social interaction—warmth, competence, and discomfort- in a concise questionnaire. In our studies, each visitor will complete the RoSAS survey immediately after their tour and take a brief questionnaire on the core DPI concepts they encountered and give an overall satisfaction rating. This approach ensures consistency with CSSR4Africa's methodology and yields clear, quantitative insights into the added value of our adaptations.

Project Work Plan

The project work plan is summarized in the Gantt chart in Figure 2. Detailed work package descriptions are provided below, along with a list of deliverables in Table 2, a list of milestones in Table 3, risk mitigation strategies in Table 4, and a summary of effort by partner in Table 5.

Collaboration

Effective collaboration between the team in this spin-off project and the CSSR4Africa team will achieved by several means, e.g., daily communication using emails, weekly physical meetings, and monitoring of weekly progress reports. Research associates will be encouraged to keep the work in sync and not work redundant tasks.

2.4 Work Package Descriptions

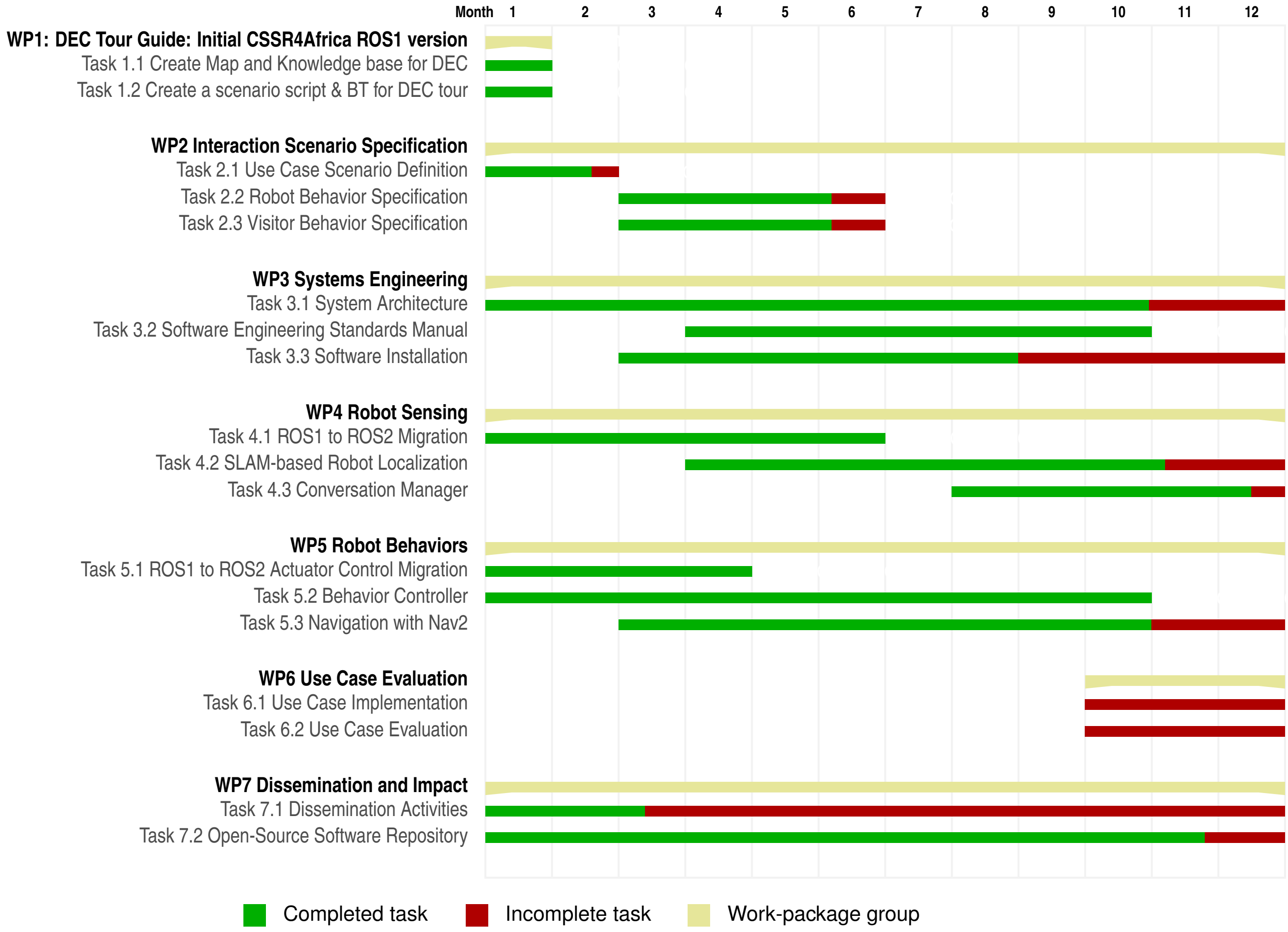

Figure 2.2: Gantt diagram showing the tasks in each work package.

Work Package Number	1
Work Package Title	DEC Tour Guide: Initial CSSR4Africa ROS1 version

Objectives

1. Adapt the current version of the CSSR4Africa software to the DEC.
2. Create a Map of the DEC.
3. Create a scenario script and control pepper to give a tour.

Description of Work

Task 1.1: Create Map and Knowledge base for DEC (M1)
The objective of this task is to create a map of the digital experience center. A map containing the booths and obstacles in the space, to facilitate navigation by the robot. This task also involves creating a knowledge base of the location of the booths (and information about the projects under it) that would be included in the tour.

Task 1.2: Create a scenario script and behavior tree for DEC tour (M1)
The objective of this task is to break down the requirements for a tour of the DEC into actionable elements. These will be organized in a sequence of events that constitute a tour of the DEC in the form of a scenario script. This scenario will then be implemented into a behavior tree specification that can be interpreted by the script interpreter developed in CSSR4Africa.

List of Deliverables

Number	Deliverable Title	Responsible Partner	Delivery Month
D1.1	DEC Map and Knowledge Base		M1
D1.2	DEC Tour Scenario Script and Behavior Tree		M1

Description of Deliverables

D1.1 DEC Map and Knowledge Base
Deliverable type: map and report.
This deliverable represents the outcome of Task 1.1. It comprises a map of the Digital Experience Center containing the booths and obstacles in the space, along with a knowledge base of the location of the booths and information about the projects featured in the tour.

D1.2 DEC Tour Scenario Script and Behavior Tree
Deliverable type: report and software.
This deliverable represents the outcome of Task 1.2. It presents a detailed scenario script that breaks down the requirements for a tour of the DEC into a sequence of events, implemented as a behavior tree specification interpretable by the CSSR4Africa script interpreter.

Work Package Number	2
Work Package Title	Interaction Scenario Specification

Objectives

1. To develop the use case scenario — DEC tour guide — and define detailed procedures for their execution.
2. To specify in detail the desired robot behavior from the point of view of the robot's manager, i.e., the person responsible for assigning the role of DEC tour guide to the robot.
3. To specify in detail the expected visitor behavior, i.e., the expected interactions by the person to whom the tour is being given. As noted already, this project does not address the interpretation of natural non-verbal human communication such as gestures and emotions. Automated speech recognition will be investigated as the main form of human interaction.

Description of Work

Task 2.1: Use Case Scenario Definition (M1–M2)
The objective of this task is to define the use case: DEC tour guide. This objective will be achieved by detailing the aim of the use-case scenario, the setting, and the procedure: a step-by-step implementation of the scenario. The implementation will be based on a structured walk-through of all the interactions that instantiate the scenario. The purpose of the walk-through is to unwrap the interaction in the use case into micro-steps of elementary perceptions as seen by the robot (from the perspective of the robot's manager) and actions by the interaction partner, i.e., the visitor. Once this timeline of elemental perceptions and actions has been unwrapped, we can then identify the measurable sensory indicators required to parameterize and quantify the information about visitor that is necessary to allow the robot to interact effectively (i.e., in a culturally sensitive manner) with her or him, e.g., locating the position of the visitor, their face and gaze. As noted already, the main form of human interaction will be through spoken requests and instructions, implemented with automated speech recognition. This unwrapped walk-through provides the baseline data for tasks T2.2 and T2.3. The outcome of this task is described in Deliverable D2.1.

Task 2.2: Robot Behavior Specification (M3–M6)
The objective of this task is to generate a list of ROS2 nodes that will implement desired robot behaviors, i.e., sequences of actions, that the gesture, speech, and navigation subsystem in WP5 will later synthesize. This objective will be achieved by analysing the use case scenario defined in Deliverable D2.1. The outcome of this task is documented in Deliverable 2.2.

Task 2.3: Visitor Behavior Specification (M3–M6)
The objective of this task is to generate a list of ROS2 nodes that will implement the perceptual functions that will enable the robot to interact effectively with the visitor and interpret the visitor behaviors specified in the use case scenarios. This project does not address the interpretation of natural non-verbal human communication such as gestures and emotions. The main form of interaction by the human will be through automated speech recognition. Thus, the robot must be able to address the visitor, and this requires certain perceptual capabilities. Hence, the objective of this task is to identify the ROS2 nodes that will provide these capabilities. This objective will be achieved by analysing the use case scenario defined in Deliverable D2.1. The outcome of this task is documented in Deliverable D2.3.

List of Deliverables

Number	Deliverable Title	Responsible Partner	Delivery Month
D2.1	Use Case Scenario Definition		M2
D2.2	Robot Behavior Specification		M6
D2.3	Visitor Behavior Specification		M6

Description of Deliverables

D2.1 Use Case Scenario Definition

Deliverable type: report.

This deliverable represents the outcome of Task 2.1. It presents a detailed scenario definition for the use case: DEC tour guide. It describes the aim of the use case, the setting, the procedure, and the measurable variables. It provides the basis for the robot behavior specification (D2.2) and the visitor behavior specification (D2.3). The report includes a walk-through of the scenario, providing a decomposition into a time sequence of elementary robot actions. For each action, the deliverable will specify the following.

1. The set of triggers for the action, e.g., input from the robot's manager or visitor using speech input, or movements of the visitor.
2. The sensory cues that characterize each trigger.
3. The exact sequence of movements, expressions, or vocal output that constitute the robot action and their associated sensory cues.
4. The goal of the action, i.e., the expected change in the environment, the response of the visitor, or the robot's manager.
5. The sensory cues that characterize the goal of the action; in each case, there may be multiple triggers and responses.

The report will also detail the layout of the environment in which the scenario is set.

D2.2 Robot Behavior Specification

Deliverable type: report.

This deliverable represents the outcome of Task 2.2. It identifies the robot's behaviour in the use-case scenario described in Deliverable D2.1 Use Case Scenario Definition, Sections 3 and 4 of which define the robot actions, component movements and sensory cues, and the associated sensory-motor process specified in the use case interactions.

The purpose of this deliverable is simply to identify the ROS2 nodes that will implement these actions. It also identifies the ROS2 packages in which the ROS2 nodes will be implemented. The detailed specification of these nodes will be provided in Deliverable D3.1 System Architecture.

The dynamics of the interaction between the visitor and the robot will be executed by Deliverable D5.2 Behavior Controller.

Description of Deliverables (continued)

D2.3 Visitor Behavior Specification

Deliverable type: report.

This deliverable represents the outcome of Task 2.3. It identifies the visitor's behaviour in the use-case scenario described in Deliverable D2.1 Use Case Scenario Definition, Sections 3 and 4 of which define the visitor actions and describe them in perceptual terms, from the perspective of the robot.

The purpose of this deliverable is simply to identify the ROS2 nodes that will provide this sensory functionality. It also identifies the ROS2 packages in which the ROS2 nodes will be implemented. The detailed specification of these nodes will be provided in Deliverable D3.1 System Architecture.

The dynamics of the interaction between the visitor and the robot will be executed by Deliverable D5.2 Behavior Controller.

Work Package Number	3
Work Package Title	Systems Engineering

Objectives

1. Design a software architecture that will facilitate the integration of the results of the project into a complete operational culturally sensitive social robotics system.

2. Formalize the integration process by identifying appropriate software engineering standards and quality assurance procedures.

3. Carry out this integration and quality assurance process.

Description of Work

Task 3.1: System Architecture (M1–M12)
The objective of this task is to design the system architecture to integrate the functionality developed in work packages WP4 and WP5, providing the necessary infrastructure to implement the use case, including domain- and task-specific knowledge. This objective will be achieved by modifying the system architecture for CSSR4Africa to accommodate for the additional subsystem, conversation manager, and for changes made to other subsystems like robot localization. The task is guided by the requirements encapsulated in Deliverables D2.1 User Case Scenario Definition, D2.2 Robot Behavior Specification, and D2.3 Visitor Behavior Specification. Architecturally, the software developed in work packages WP4 and WP5 will be viewed as modular subsystems. The goal of this task is to specify the functionality of each of these subsystems and to define their interfaces, i.e., the manner in which data and control signals are input to and output from these subsystems. These interfaces will be specified at both a high level of abstraction (e.g., what data is input and output and how it is represented) and at a low-level of abstraction (how this data is exposed through subsystems interfaces and how the functionality of the subsystem can be externally configured by other subsystems). The low-level interface and functional configuration facilities will be effected using the ROS2 (Robot Operating System). Interface design and module specification will be done in collaboration with the respective module development tasks in work packages WP4 and WP5. The outcome of this task is described in Deliverable D3.1.

Task 3.2: Software Engineering Standards Manual (M1–M12)
The objective of this task is to define a set of project standards governing the specification, design, documentation, and test of all software to be developed in work packages WP4 and WP5. The specification standards will address functional definition, data representation, and module / sub-system behavior. Design standards will focus on the decoupling of functional computation, module communication, external module configuration, and inter-module coordination. To maximize compositionality, a best-practice component-based software engineering model. The outcome of this task is described in Deliverable D3.2.

Task 3.3: Software Installation (M3–M12)
The objective of this task is to document the process for the installation and execution of the software required to instantiate all or part of the system and run the unit, integration, and system tests, and the use case scenario on the physical robot. This objective will be achieved by documenting the installation of all software components on an ongoing basis, as a living document, to ensure it reflects the current capabilities of the system. The outcome of this task is described in Deliverable D3.3.

List of Deliverables

Number	Deliverable Title	Responsible Partner	Delivery Month
D3.1	System Architecture		M12
D3.2	Software Installation Manual		M12
D3.3	System Integration and Quality Assurance		M12

Description of Deliverables

D3.1 System Architecture
Deliverable type: report.
This deliverable represents the outcome of Task 3.1. Describes the system architecture in detail, identifying the component subsystems, the modules comprising each subsystem, and the information exchanged between subsystems and modules, including a specification of the data that are input to each module, the data that are output from each module, and the data that are used to control the operation of the module, including the manner in which these data are made available or accessed, either through ROS2 topics, services, or actions. Specifically, the deliverable provides the functional specification of the two subsystems to be developed in the WP4 robot sensing work package and the subsystems to be developed in the WP5 robot behaviors work package. All data will be specified by both the information content and the representation.

D3.2 Software Engineering Standards Manual
Deliverable type: software and report.
This deliverable represents the outcome of Task 3.2. It comprises functionally integrated software at different stages of the system development. In addition to functional code, the deliverable will include a report on the results of integration tests. The deliverable will be issued at the end of the project. The final version will present the final system test and a test report, along with a system user manual and a system reference manual.

D3.3 Software Installation Manual
Deliverable type: report.
This deliverable represents the outcome of Task 3.3. It documents the process for the installation and execution of the software required to instantiate all or part of the system and run the unit, integration, and system tests, and the case scenario on the physical robot. It will be created as a living document to ensure it reflects the current capabilities of the system.

Work Package Number	4
Work Package Title	Robot Sensing

Objectives

1. Develop a suite of unit tests to verify that all sensors are functioning correctly and that the sensor data can be accessed using ROS2.
2. Migrate the nodes from CSSR4Africa that will detect and localize people, including their face and gaze, localize sounds, and localize the robot in a world frame of reference to ROS2.
3. Develop conversation manager to generate responses to prompts by an interaction partner.
4. Migrate Speech Event, which performs automated speech recognition, to ROS2.

Description of Work

Task 4.1: ROS1 to ROS2 Migration (M1–M6)
The objective of this task is to migrate the Work Package 4 software modules, originally developed in ROS1 under the CSSR4Africa work plan (`https://cssr4africa.github.io/workplan`), to ROS2 (Humble) while enhancing their perceptual capabilities for culturally sensitive human-robot interaction.

This objective will be achieved by porting the Person Detection and Localization, Face & Mutual Gaze Detection and Localization, Sound Detection and Localization, and Speech Event nodes from ROS1 to ROS2, ensuring full compatibility with the ROS2 framework. Beyond functional equivalence with the original implementations, the migration will incorporate additional functionality — such as age and gender detection from facial imagery — to enrich the robot's sensing capabilities. These additional cultural sensitivity cues will enable the robot to adapt its behaviour and communication style by enhancing the overall quality and appropriateness of the interaction.

Task 4.2: SLAM-based Robot Localization (M3–M8)
The objective of this task is to develop a software module to determine the pose (position and orientation) of the robot in a Cartesian world frame of reference.

This objective will be achieved by a combination of relative position estimation and SLAM-based localization. For relative position estimation, both odometry and the use of the robot's inertial management unit will be investigated. For absolute position estimation, we will investigate SLAM solutions, RTAB-Map or GMapping. To estimate the orientation of the robot, we will ignore adjustments of body posture through rotation about the X- and Y-axes, and we will restrict estimation to rotation about the robot's Z-axis in the based frame of reference, i.e., rotation about the Z-axis. The software development process will involve requirements definition, module specification, interface design, module design, coding, and unit testing. The outputs of each of these phases are detailed in Deliverable D4.2.

Description of Work (continued)

Task 4.3: Conversation Manager (M3–M6)
The objective of this task is to configure, test, and deploy a conversation manager on the Pepper robot for generating responses to prompts by an interaction partner in the language of English. This will enable the robot to have a meaningful conversation with visitors in the context of the DEC in particular, and CMU-Africa in general.

This objective will be achieved by making use of open-source LLMs and prompt engineering to the purpose of a tour of the DEC. The conversation manager will use transcribed text from the module developed Speech Event as prompts to generate a response. The output of the LM in this module may also be a classification of the prompt as a comment, question, or instruction, this would allow the robot to take any further necessary actions in addition to responding to the visitor in speech.

List of Deliverables

Number	Deliverable Title	Responsible Partner	Delivery Month
D4.1	ROS1 to ROS2 Migration		M6
D4.2	SLAM-based Robot Localization		M8
D4.3	Conversation Manager		M6

Description of Deliverables

D4.1 ROS1 to ROS2 Migration
Deliverable type: software and report
This deliverable represents the outcome of Task 4.1. It comprises the documented software required to build and launch the modules migrated to ROS2 as individual ROS nodes that encapsulates the functionalities described in the CSSR4Africa work plan. The ROS nodes will have the same names as before.

In addition to functional code, the deliverable will include a report with sections detailing the outputs of each phase of the migration process.

D4.2 SLAM-based Robot Localization
Deliverable type: software and report
This deliverable represents the outcome of Task 4.2. It comprises the documented software required to build and launch a module implemented as a single ROS node to determine the pose (position and orientation) of the robot in a Cartesian world frame of reference.

In addition to functional code, the deliverable will include a report with sections detailing the outputs of each phase of the software development process.

The requirements definition section will specify the functional needs of users of the module. This will involve a review and update of the requirements as set out in the task description, identifying gaps and misalignments with actual needs.

The module specification section will define the functional characteristics as follows. This module will compute the pose of the robot in the world frame of reference. It will do this continuously, in real time, by updating the current pose based on relative pose estimation, using either odometry or the robot's inertial management unit IMU (or a combination of both). Since pose estimation errors using relative techniques grow with time, the module will periodically reset its pose estimate using absolute pose estimation. Position estimation will be accomplished by SLAM-based solutions.

For relative pose estimation, the input will be the odometry data published by the robot and data from the robot's accelerometer and gyrometer. For absolute pose estimation, the input will take the form of an RGB image from one of the robot's cameras and a depth image from one of the robot's depth sensors, and a LiDAR sensor. The module can also provide service requests to reset the robot's pose using absolute pose estimation.

The module can run in normal mode or verbose mode. In verbose mode, data that is published to topics is also printed to the terminal. Also, if the module is operating in verbose mode, the RGB image will be displayed in an OpenCV window.

Description of Deliverable (continued)

The names of the topics to be used for each sensor will be read from a data file comprising a sequence of key-value pairs. The key is the name of the sensor. The value is the topic name.

The operation of the module will be determined by parameters provided in a configuration file that contains a list of key-value pairs. One key-value pair will specify the RGB camera to be used (e.g., camera FrontCamera | RealSense). One key-value pair will specify the filename of the file in which the physical Pepper robot sensor and actuator topic names are stored (e.g., robotTopics pepperTopics.yaml). One key-value pair will specify whether diagnostic data is to be printed to the terminal and diagnostic images are to be displayed in OpenCV windows (e.g., verboseMode true). The configuration file will be named robotLocalizationConfiguration.yaml.

The interface design section will include a specification of the data that are input to the module, the data that are output from the module, and the data that are used to control the operation of the module, including the manner in which this data is made available or accessed, either through ROS topics, services, or actions.

The module design section will specify selected algorithms and data structure. It is planned to consider two options, i.e., RTAB-Map and GMapping.

The coding section will contain the functional program code with internal documentation, any required build files, driver program code, and stub program code. Code will be written and documented in adherence to the standards to be developed in Deliverable D3.2 Software Engineering Standards Manual.

In addition, a user manual will be included in the deliverable report.

Description of Deliverables (continued)

D4.3 Conversation Manager
Deliverable type: software and report
This deliverable represents the outcome of Task 4.3.2. It comprises the documented software required to build and launch a module implemented as a single ROS node that processes the prompts, spoken by an interaction partner in the English language, and generates relevant responses in written text. The ROS node will be named conversation_manager.

In addition to functional code, the deliverable will include a report with sections detailing the outputs of each phase of the software development process.

The requirements definition section will specify the functional needs of the module users. This will involve a review and update of the requirements as set out in this work plan, identifying gaps and misalignment with actual needs.

The module specification section will specify the functional characteristics, detailing the input-to-output data transformation, expected input data, expected output data, module configuration parameters, and a draft user manual. Functionally, this module first requires that an LLM be implemented on-premise, trained or augmented, and tested so that it can perform generate accurate responses to user prompts. This model will subsequently be used to take as input the transcribed text of a visitor's utterance by the module developed in Task 4.3.1 Speech Event, and produce the corresponding response.

The module can run in normal or verbose mode. In verbose mode, the menu and response are also printed on the terminal. The operation of the module will be determined by the parameters provided in a configuration file. These parameters comprise a list of key-value pairs. One key-value pair will specify whether diagnostic data are to be printed on the terminal and diagnostic images are to be displayed in OpenCV windows (e.g., verboseMode true).

The input will take the form of a text from speechEvent module, published on a topic named /speechEvent/text.

The output will take the form of a record containing a string that represents the appropriate response in the context of the conversation. This will be published on a topic named /prompt. The module can run in normal mode or verbose mode. In verbose mode, data that is published to topics is also printed to the terminal.

The interface design section will include a specification of the data that are input to the module, the data that are output from the module, and the data that are used to control the operation of the module, including the manner in which these data are made available or accessed, either through ROS topics, services, or actions.

Description of Deliverables (continued)

The module design section will comprise a complete and comprehensive description of the conversation manager and its configuration procedure.

The coding section will contain the functional program code with internal documentation and any required build files. Code will be written and documented in adherence to the standards to be developed in Deliverable D3.2 Software Engineering Standards Manual.

In addition, a user manual will be included in the deliverable.

Work Package Number	5
Work Package Title	Robot Behaviors

Objectives

1. Migrate Pepper's actuator control system from ROS1 to ROS2 Humble, ensuring full compatibility with the ROS2 control framework while maintaining existing functionality and performance standards through comprehensive testing and validation.

2. Implement an interaction manager that sequences interaction between the robot and the visitor using a ROS2-compatible behavior tree or state machine framework, integrating the cultural ontology into the mission interpreter for contextually appropriate interactions.

3. Replace the baseline navigation approach with the Nav2 stack on ROS2, utilizing the SLAM-generated map of the DEC layout for robust global planning and local obstacle avoidance, specifically tuned for Pepper's footprint and sensor configuration.

4. Develop and implement a gesture animation system that enables Pepper to perform culturally appropriate gestures and expressions, utilizing ROS2 action servers for coordinated sound and motion timing.

Description of Work

Task 5.1: ROS1 to ROS2 Actuator Control Migration (M1–M4)
The objective of this task is to migrate Pepper's existing actuator control system from ROS1 to ROS2 Humble, ensuring full compatibility with the ROS2 control framework while maintaining existing functionality and performance standards. This migration forms the foundation for all subsequent behavioral implementations.

The following nodes from CSSR4Africa will be migrated Task (5.2) Animate Behavior Subsystem, Task (5.3) Attention Subsystem, Task(5.4.3) Robot Mission Interpreter, Task (5.5.1) Gesture Execution, Task (5.5.2.4), and Integrated Text-to-Speech Conversion.

The migration will begin with a comprehensive review of Pepper's existing ROS1 actuator control architecture, identifying all hardware interface plugins, controller configurations, and motion primitive implementations. The ROS2 control framework introduces significant architectural changes, including the controller manager lifecycle, hardware abstraction layers, and standardized interfaces that must be properly implemented for Pepper's specific joint configuration. Three key components will be addressed: hardware interface plugins that communicate with Pepper's 20+ joint controllers (including head, arms, hands, hip, and knee joints), controller configurations that define Pepper's motion behaviors and gesture primitives, and the transition mechanisms that ensure smooth operation within the ROS2 control framework. Each component will be systematically ported, tested, and validated against the original ROS1 implementation to ensure functional equivalence across Pepper's full range of motion.

As an enhancement to the migrated Gesture Execution node, the system will incorporate speech-gesture synchronization to enable Pepper to perform natural, culturally appropriate gestures during speech delivery. This will be achieved through a keyword-based gesture mapping approach, where specific phrases trigger corresponding hand pointing, welcoming gestures, or demonstrative movements. The gesture animation capability will be integrated directly into the Gesture Execution node rather than implemented as a separate task.

Description of Work (continued)

A compatibility testing framework will be established to validate that migrated controllers produce identical motion outputs to their ROS1 counterparts. This will include joint trajectory execution tests and verification of velocity and acceleration profiles. The testing framework will utilize hardware validation on the physical Pepper robot; the simulation environment will not be migrated.

Task 5.2: Behavior Controller (M1–M10)
The objective of this task is to implement an interaction manager that sequences role-play states using a ROS2-compatible behavior tree or state machine framework, integrating the cultural ontology into the mission interpreter for culturally appropriate interactions.

The interaction manager will be designed as a hierarchical state machine that can handle complex interaction sequences while maintaining flexibility for dynamic context changes. The system will integrate the cultural ontology developed in previous work packages to ensure that all interactions are culturally appropriate and contextually relevant. Then it will use the outcome of Task 2.1, documented in Deliverable D2.1, by recruiting the robot and visitor behavior documented in Deliverables D2.2 and D2.3. It will also use the cultural knowledge ontology and cultural knowledge base developed in the CSSR4Africa Project (Task 5.4.1) `https://cssr4africa.github.io/deliverables/CSSR4Africa_Deliverable_D5.4.1.pdf`

Role-play state sequencing will be implemented using a modular approach, where each role-play scenario is defined as a separate behavior tree that can be dynamically loaded and executed. This approach allows for easy addition of new scenarios and modification of existing ones without requiring system rebuilds.

The behavior tree framework will be selected based on ROS2 compatibility, performance characteristics, and extensibility requirements. Candidate frameworks include BehaviorTree.CPP, and Groot. The selected framework will be extended with DEC-specific nodes that can query the cultural ontology and make context-aware decisions. Mission interpretation will be enhanced with natural language processing capabilities that can parse user requests and map them to appropriate role-play scenarios. The system will support English language primarily, and maybe Kinyarwanda if time permits, and cultural communication styles, adapting its response patterns based on user preferences and cultural background.

Task 5.3: Navigation with Nav2 (M1–M10)
The objective of this task is to replace the baseline navigation approach with the Nav2 stack on ROS2, utilizing the SLAM-generated map of the DEC layout for robust global planning and local obstacle avoidance. The system will be specifically tuned for Pepper's footprint and sensor configuration.

The SLAM-generated map will be integrated into the Nav2 framework, with appropriate map processing to handle dynamic obstacles and environmental changes. Static map layers will be augmented with dynamic obstacle information from Pepper's sensors, creating a comprehensive environmental representation suitable for navigation planning. Global planning will utilize A* or similar algorithms optimized for the DEC layout, taking into account high-traffic areas and cultural spaces that require respectful navigation. Local planning will implement dynamic window approaches or similar techniques that can handle real-time obstacle avoidance while maintaining smooth motion profiles.

Description of Work (continued)

Parameter tuning will be conducted through systematic testing in both simulation and real-world environments. Key parameters include planner timeouts, obstacle inflation radii, trajectory scoring weights, and recovery behavior triggers. The tuning process will optimize for safety, efficiency, and user comfort in the DEC environment.

List of Deliverables

Number	Deliverable Title	Responsible Partner	Delivery Month
D5.1	ROS2 Actuator Control System		M4
D5.2	Behavior Controller		M10
D5.3	Advanced Nav2 Navigation System		M10

Description of Deliverables

D5.1 ROS1 to ROS2 Actuator Control Migration
Deliverable type: software and report
This deliverable represents the outcome of Task 5.1. It comprises the documented software required to build and launch the modules migrated to ROS2 as individual ROS nodes that encapsulate the functionalities described in the CSSR4Africa work plan. The ROS nodes will have the same names as before.

In addition to functional code, the deliverable will include a report with sections detailing the outputs of each phase of the migration process.

D5.2 Behavior Controller
Deliverable type: Report
This deliverable represents the outcome of Task 5.3. It comprises the documented software required to develop and implement the Behavior Controller, which uses XML-based mission definitions to sequence robot interactions for execution within the CSSR4Africa Project framework. The system will create structured XML mission definitions that specify interaction sequences and behavioral coordination.

The requirements definition section will specify the functional needs for adapting the CSSR4Africa Robot Mission Interpreter to the DEC environment, including role-play scenario execution, cultural sensitivity integration, and dynamic mission adaptation based on user interactions.

The Behavior Controller XML schema will support mission elements including interaction sequences, cultural ontology references, speech synthesis commands, and navigation waypoints. Mission definitions will incorporate cultural parameters and behavioral requirements while maintaining compatibility with the CSSR4Africa behavior controller's parsing capabilities. The Behavior Controller will support XML templates for common interaction patterns, reusable cultural behavior components, and modular mission segments that can be combined to create complex interaction scenarios. Template instantiation will allow for rapid mission development while maintaining consistency.

D5.3 Nav2 Navigation System
Deliverable type: software and report
This deliverable represents the outcome of Task 5.3. It comprises the documented software required to build and launch a navigation system based on the Nav2 stack, specifically configured and tuned for Pepper's physical characteristics and the DEC environment layout. The system will be implemented as a collection of coordinated ROS2 nodes following the Nav2 architecture.

Description of Deliverables (continued)

In addition to functional code, the deliverable will include a report with sections detailing the outputs of each phase of the software development process.

The requirements definition section will specify the navigation needs for Pepper operating in the DEC environment, including obstacle avoidance, cultural space navigation, accessibility compliance, and integration with interaction behaviors.

The module specification section will define the Nav2 stack configuration optimized for Pepper's omnidirectional base, sensor configuration, and physical footprint. The system will utilize SLAM-generated maps of the DEC layout with dynamic obstacle integration from Pepper's sensor suite including cameras, laser scanner, and sonar sensors.

Global planning will implement path planning algorithms optimized for the DEC environment, culturally sensitive spaces requiring respectful navigation approaches, and accessibility requirements. Local planning will provide real-time obstacle avoidance with smooth trajectory generation suitable for Pepper's movement characteristics. The navigation system will implement lifecycle management for robust operation, including proper initialization sequences, error recovery mechanisms, and graceful shutdown procedures. Recovery behaviors will be specifically tuned for Pepper's capabilities and the DEC environment constraints.

Parameter configuration will be extensively tuned for Pepper's physical properties including footprint dimensions, velocity limits, acceleration constraints, and sensor characteristics. Obstacle inflation parameters, trajectory scoring weights, and planning timeouts will be optimized through systematic testing. Configuration modes will include standard navigation, cultural-sensitive navigation with reduced speeds in designated areas, and enhanced obstacle detection and alternative route planning.

The module design section will detail Nav2 component configurations, parameter tuning methodologies, map integration procedures, and coordination service implementations.

Unit testing will validate navigation accuracy, obstacle avoidance performance, cultural space compliance, and coordination with other robot subsystems. Testing will include both simulation validation and physical robot trials in the DEC environment.

The user manual will explain navigation goal setting, map updating procedures, parameter adjustment for different environments, and troubleshooting common navigation issues specific to Pepper and the DEC layout.

The coding section will contain functional program code with internal documentation, build files for ROS2 compilation, and integration code for speech synthesis interfaces. Code will adhere to the standards developed in Deliverable D3.2 Software Engineering Standards Manual.

Description of Deliverables (continued)

The unit testing section will present tests for each synchronization approach on both physical robot and simulator platforms. Tests will verify gesture-speech timing accuracy, cultural appropriateness validation, and actuator safety constraint compliance. Launch files will configure the required ROS2 nodes and select the appropriate platform configuration.

A user manual will explain how to configure gesture libraries, select synchronization approaches, adjust cultural parameters, and integrate the system with speech synthesis and interaction management components.

Work Package Number	6
Work Package Title	Use Case Demonstration and Evaluation

Objectives

1. Carry out initial demonstrations of the system architecture in the DEC tour use case.
2. Evaluate the success of the demonstrations and identify any adjustments to the output from work packages WP1 - WP5.
3. Carry out final demonstrations.
4. Re-evaluate the success of the demonstrations.

Description of Work

Task 6.1: Use Case Implementation (M7–M8)
The objective of this task is to implement both use cases using the outcomes of WP1 - WP5, i.e., the cultural knowledge, the scenario specification, and the integrated robot's sensory and interaction capabilities. This objective will be achieved by running the unit test and system tests in Deliverable D3.3. The outcome of this task is described in Deliverable D6.1.

Task 6.2: Use Case Evaluation (M9–M12)
The objective of this task is to evaluate the implementation using user feedback and observations from functional demonstrations and produce a set of required adjustments for the interaction primitives and design patterns. The task uses the outcome of Task 6.1 as documented in Deliverable D6.1. The outcome of this task is described in Deliverable D6.2.

List of Deliverables

Number	Deliverable Title	Responsible Partner	Delivery Month
D6.1	Use Case Implementation		M8
D6.2	Use Case Evaluation		M12

Description of Deliverables

D6.1 Use Case Implementation
Deliverable type: demonstration and report
This deliverable represents the outcome of Task 6.1. It is a demonstration of the complete working system for the use case defined in Work Package WP2.

D6.2 Use Case Evaluation
Deliverable type: report
This deliverable represents the outcome of Task 6.2. It will evaluate and document the success of the initial use case implementation by showing demonstrations to some visitors and getting feedback.

Work Package Number	7
Work Package Title	Dissemination and Impact

Objectives

1. Create a project website.
2. Disseminate the ongoing status of the project in several forums, including the project website.
3. Publish research results in conferences and journals.
4. Make the project software and data freely available online.

Description of Work

Task 7.1: Dissemination Activities (M1–M12)
The objective of this task is to effect scientific dissemination through leading international conferences and journals, as well as within the Upanzi network team and the broader CMU-Africa community. A record of all communication actions will be kept and incorporated in an end of project dissemination and communication report. The outcome of this task is described in Deliverable D7.2.

Task 7.2: Open-Source Software Repository (M9–M12)
The objective of this task is to make software, data, and protocols publicly available on GitHub and maintain the repositories. The outcome of this task is described in Deliverable D7.3.

List of Deliverables

Number	Deliverable Title	Responsible Partner	Delivery Month
D7.1	Dissemination Activities		M12
D7.2	Open-Source Software Repository		M12

Description of Deliverables

D7.1 Dissemination Activities
Deliverable type: report
This deliverable represents the outcome of Task 7.1. It documents the reports, papers, articles, presentations, and demonstrations that have been carried out.

D7.2 Open-Source Software, Data, and Designs
Deliverable type: online repository and report
This deliverable represents the outcome of Task 7.2. It is a GitHub software repository of the software and data (i.e., the cultural knowledge) developed in the project. The initial repository will be created by the first month and will be updated as software is developed throughout the duration of the project.

Table 3: List and Schedule of Milestones

Number	Milestone Title	Related WP No.	Month	Means of Verification
M1	DEC tour guide initial CSSR4Africa ROS1 version complete	WP1	1	D1.1, D1.2
M2	System architecture and standards complete	WP3	1	D3.1, D3.2
M3	ROS2 migration and core behavior implementation complete	WP3, WP4, WP5	6	D3.3
M4	Use case demonstration and evaluation complete	WP6	12	D6.2

References

Bartneck, C., Belpaeme, T., Eyssel, F., Kanda, T., Keijsers, M., & Sabanovic, S. (2020). *Human-robot interaction – an introduction.* Cambridge University Press.

Brugali, D., & Scandurra, P. (2009, December). Component-Based Robotic Engineering (Part I). *IEEE Robotics and Automation Magazine*, 84–96.

Brugali, D., & Shakhimardanov, A. (2010, March). Component-Based Robotic Engineering (Part II). *IEEE Robotics and Automation Magazine*, 100–112.

Heineman, G. T., & Council, W. T. (2001). *Component-Based Software Egineering: Putting the pieces Together.* Reading, Massachusetts: Addison-Wesley.

Kahn, P. H., Freier, N. G., Kanda, T., Ishiguro, H., Ruckert, J. H., Severson, R. L., & Kane, S. K. (2008). Design patterns for sociality in human-robot interaction. In *Proc. 3rd ACM/IEEE international conference on human-robot interaction* (pp. 97 – 104).

Szyperski, C. (2002). *Component software: Beyond object-oriented programming.* Reading, Massachusetts: Addison-Wesley.

Vernon, D., Billing, E., Hemeren, P., Thill, S., & Ziemke, T. (2015). An architecture-oriented approach to system integration in collaborative robotics research projects - an experience report. *Journal of Software Engineering for Robotics*, *6*(1), 15–32.

Document History

Version 1.0
First draft.
Muhammed Danso and Yohannes Haile.
14 July 2025.

Version 1.1
Updated the Gantt chart
Muhammed Danso and Yohannes Haile.
21 January 2026.

Version 1.2
Changed the name from language model to conversation manager.
Updated the Robot Mission Language to Behavior Controller.
Yohannes Haile.
04 March 2026.

Version 1.3
Removed the gesture execution animation. Now it will be part of the text to speech model.
The report will be included in Task 4.1.
Yohannes Haile.
15 March 2026.

Version 1.4
Updated to change the software Quality Assurance to Software Engineering Standard.
Removed the migration of Actuator test from the list of the ROS1 to ROS2 migration list.
Removed Advanced from the naming of the Navigation task in Task 5.3.
Fixed the months in the description of work.
Yohannes Haile.
02 April 2026.

Version 1.4
Updates to the Gantt Chart.
Yohannes Haile.
08 May 2026.

Version 1.5
Removed Task 1.1 “Review Rwandan Cultural Knowledge Survey” and Task 1.2 “Use Case Feedback” from the African Cultural Knowledge work package section as these are no longer needed.
Yohannes Haile.
01 June 2026.

Version 1.6

Added Picture of the DEC Center.
Removed all references to driver and unit testing.
Update the topics and node name for the conversation manager.
Updated the milestone pages.
Yohannes Haile.
02 June 2026.