**Team : 09**

**Autonomous Mobile Robots**

**Adaption Phase:**

|  |  |
| --- | --- |
| S1: System Configuration and calibration | The Robot is installed in the warehouse and a Virtual map is created by moving the Bot around the warehouse. |
| S2: Configuration of agent with Environment | The environmental details like Rack location, height, width, product locations are to be updated and configured with Bot. |
| S3: Installing Charge Docking Stations | Charge Dockers are to be installed in the warehouse and the Agent is provided with data. |

**Operation Phase:**

|  |  |
| --- | --- |
| S4: System Start and self-check | The bot should be powered and it has to go through self-diagnosis and update it to the database and notifies the operator informing its is in the ready state. |
| S5: Receiving Tasks | Bot should Receive the Tasks by the operator. the commands consists of part collecting location and part Delivering Locations. Bot should update the received task to database |
| S6: Planning its Task | The plan for completing the task is to be formed by the robot in such a way that it should complete the task with less time, the proper path has to be planned by the robot. |
| S7: Executing the Task | Bot should Collect the part from collecting location and starts to move, it should dynamically create the alternate path while subjected to obstacle and bot should inform other robots while executing task to avoid collision and deadlock. |
| S8: Error Handling | When bot is subjected to any errors or malfunction, it should terminate its task by raising the token to other bots to complete its task. |
| S9: Completing Task | Bot should place the part in specified Delivery location and update the status to database. |
| S10 : Validating and Updating Database | The bot should validate the task, update it to database and Change its status to Finished. |

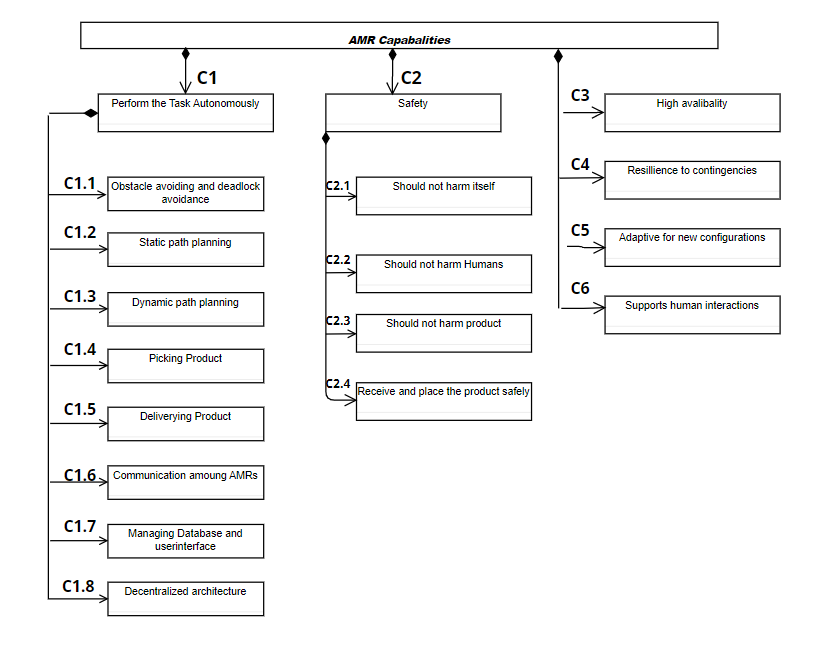
**Charging and Maintenance Phase:**

|  |  |
| --- | --- |
| S11: Error handling | Whenever bot subjected to any error, it should rise a flag update the database and rise a token to other bots to complete the executing task. |
| S12: Auto Docking | The bot should automatically dock itself to the charging station, by sensing the drop in battery. |

**Description of Scenarios:**

|  |  |
| --- | --- |
| **Operational Scenario** | **S1: System Configuration and calibration** |
| Preconditions | * Warehouse is cleared |
| Triggering event | The Bot is manually moved around the warehouse |
| Description | The bot collects the environmental data through the sensors and represents that data in the form of virtual map |
| Post conditions | * The Bot has created his virtual Map |
| **Operational Scenario** | **S2: Configuration of agent with Environment** |
| Preconditions | * All the environment objects dimensions are collected |
| Triggering event | The bot require an map which has updated environmental details |
| Description | User will manually fill up the details of the environment objects like racks and other objects |
| Post conditions | * The bot has updated the environment objects on the map |
| **Operational Scenario** | **S3: Installing Charge Docking Stations** |
| Preconditions | * Charging station with rated i/o parameters. |
| Triggering event | Bot require charging stations to charge itself. |
| Description | Charging stations with rated input output voltage – current parameters are installed manually in selected locations. |
| Post conditions | * Charging station with rated VI parameters. |
| **Operational Scenario** | **S4: System Start and self-check** |
| Preconditions | * Bot should have operatable battery health and Bot should power ON. |
| Triggering event | Bot is powered ON |
| Description | Once the bot is powered ON, The bot should check all the sensors-actuators conditions, communication status itself and it should update it to database. |
| Post conditions | * Self-check of bot is completed and bot is ready to carryout task. |
| **Operational Scenario** | **S5: Receiving Tasks** |
| Preconditions | * Self-check of Bot has to be done. * Bot should be in idle condition. |
| Triggering event | User updated the new product which has to be placed in the given location. |
| Description | Whenever the bot is in idle state, once the user sends the request to place or get the product from specified location, the bot should receive the task details and store it in local storage, update it to database along with its status and informs other bots about its tasks. |
| Post conditions | * Bot received the task from the user and updated to the database * Bot status is communicated to other Bots. |
| **Operational Scenario** | **S6: Planning its Task** |
| Preconditions | * The bot is received the Task from the user * Status of the Bot is “*working*” |
| Triggering event | The received the task and it is to plan the procedure to complete the task. |
| Description | Once the Bot receives the task, it should plan the locations (to and from), path (static path planning), required time period. |
| Post conditions | * Plan to complete the task is prepared by the bot itself. * Bot is ready to execute the task. |
| **Operational Scenario** | **S7: Executing the Task** |
| Preconditions | * The Planning of the task is done by bot itself. * Bot is ready to execute the task. |
| Triggering event | Planning to execute the task is done by the bot. |
| Description | Once the planning has been done, the bot should start executing it by driving the actuators and sensing the environment by its sensors, sensing and actuating should happen in real time. |
| Post conditions | * Bot is executing the task. |
| **Operational Scenario** | **S8: Error Handling** |
| Preconditions | * Bot executing the task * Bot is subjected to error |
| Triggering event | The subjected to error |
| Description | Whenever the bot is subjected to error, the bot should rise the token along with error, log the error and rise the token to others bots to complete the executing task, other bots which are in idle state and near to that bot should receive the token and start to execute the uncompleted task by updating database. |
| Post conditions | * The error in the bot is identified and ongoing task is being completed by other bot. |
| **Operational Scenario** | **S9: Completing Task** |
| Preconditions | * Bot is executing the task * It has moved near the destination location |
| Triggering event | Bot is moved to the destination location |
| Description | When the bot is moved near the destination location, it should recheck the location details and place the product. |
| Post conditions | * Bot is completed the task by placing the product at the specified location. |
| **Operational Scenario** | **S10: Validating and Updating Database** |
| Preconditions | * The bot is completed the task by placing the product to the specified location. |
| Triggering event | Bot completed the Task. |
| Description | When the bot place the product at the specified delivery location, it should confirm the location by its virtual map and environmental data, and update is to database. |
| Post conditions | * The bot completed the task and changes it status to “*completed*” |
| **Operational Scenario** | **S11:** **Error Handling** |
| Preconditions | * Bot should power ON * Bot should be in executing state |
| Triggering event | Bot is subjected to error |
| Description | Whenever bot subjected to any error, it should rise a flag update the database and rise a token to other bots to complete the executing task. |
| Post conditions | * Error of the bot is diagnosed. |
| **Operational Scenario** | **S12: Auto Docking** |
| Preconditions | * Bot should be in power ON state |
| Triggering event | When the battery capacity of the bot is low |
| Description | When the bot is having low battery capacity it should identify the docking stations and it should go there to dock itself for charging. |
| Post conditions | * The Bot is charged automatically. |

***SysML block definition diagram that captures the capabilities to be provided by the collaborative robot application***.



|  |  |
| --- | --- |
| ***CAPABILITY*** | ***DESCRIPTION*** |
| C1 Perform the task Autonomously | The AMR should be able to receive and deliver the product in the warehouse autonomously |
| C1.1 Obstacle avoiding and deadlock avoidance | The bot should able to avoid the hitting of the object and reduce the occurrence of the deadlock. |
| C1.2 Static path planning | The bot should plan the path when it received the to and from location, the path should be shortest with respect to time. |
| C1.3 Dynamic Path Planning | When the bot subjected to obstacle or any other error in the path, it should have ability to plan the path dynamically. |
| C1.4 Receiving Product | The bot should able to receive the product without any location errors. |
| C1.5 Delivering the Product | The bot should able to deliver the product without any location errors and damages. |
| C1.6 Communication among AMRs | The AMR should communicate among other AMRs where communication includes current location, status of the bot and other required information. |
| C1.7 Managing Database and user-interface | The bot should have ability to update the database frequently and to quickly response to the user requests. |
| C1.8 Decentralized architecture | The Bot should have decentralized control architecture. |
| C2 Safety | considering that the robot should not cause any harm. Harmless is a complex capability that requires both the realization of certain functionalities (e.g., C2.1), together with quality attributes. |
| C2.1 Should not harm itself | The Bot should carryout the task without harming itself. |
| C2.2 Should not harm Humans | Bot should have an ability to not to harm humans |
| C2.3 Should not harm product | Bot should carry the products without damaging them. |
| C2.4 Receive and place product safely | Handling product by the bot should be safe. |
| C3 High availability | The robotic system shall be available 98% of the. Increased availability is a complex capability that encompasses different functionalities and quality attributes. This capability will map to reliability requirements for the components of the AMR. |
| C4 Resilience to contingencies | A contingency is an unforeseen and undesired interaction with another actor or its environment. Examples of contingencies our robotic system must be able to manage are small deviations in the receiving and delivering of products, or minor irregularities in the locations of the product. |
| C5 Adaptive for new Configurations | The AMR should modular in such a way that it can able to have multiple types of load carrying mechanisms. |
| C6 Supports human interactions | The system will offer an interface for human operators to manage the system. |