

HW3: Mission Planning with ROS 2

Mobile Manipulation 2026 - ShanghaiTech University

Due Date: Refer to the course website

Attention

Before starting this assignment, ensure you have **ROS 2 Jazzy** installed on Ubuntu 24.04 and a basic understanding of ROS 2 nodes, topics, services, and custom interfaces. This is an **individual assignment**. See the provided `README.md` for setup, simulator usage, and evaluation instructions.

Mission Planning Overview

Mission planning involves breaking down a high-level task into executable actions. For example, fetching a cup of water requires picking up a cup, moving to a water source, filling the cup, and bringing it back. Your planner should choose suitable actions from the current state, execute them, and handle failures.

Your Task

We provide a map in the **osmAG** format [1], together with the ROS 2 packages `osm_parser` and `osm_interfaces`. The map contains rooms, passages, furniture, and objects, as shown in Figure 1.

Assignment package source: <https://robotics.shanghaitech.edu.cn/gitlab/moma-2026/resources/hw3>

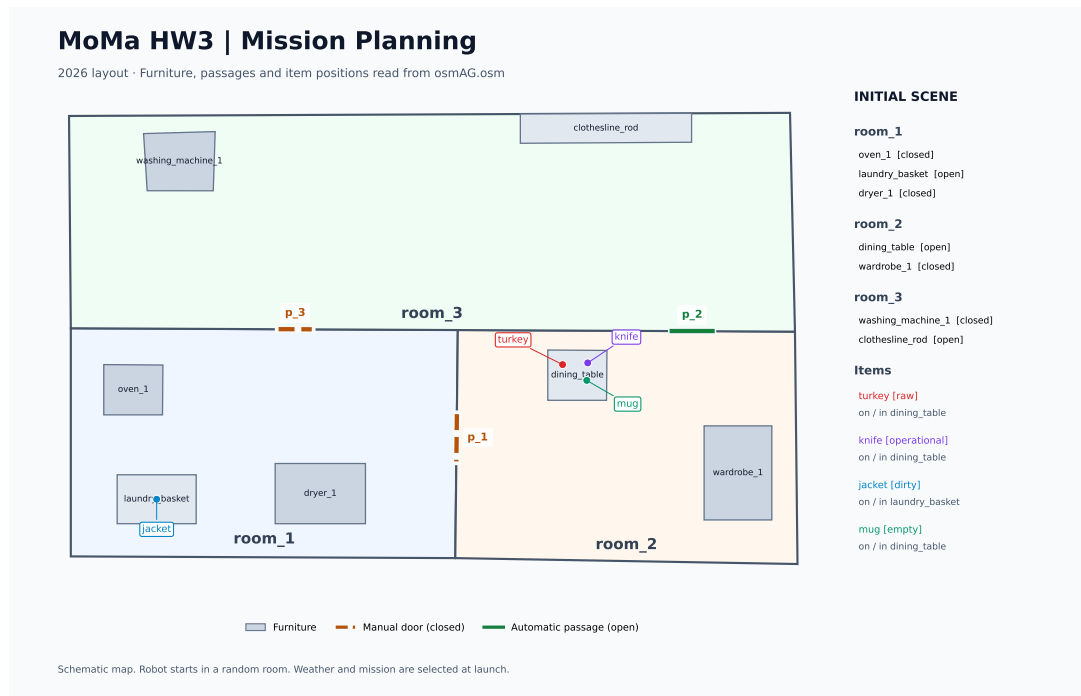


Figure 1: Map for the mission planning tasks.

Each run randomly selects one of the following missions with equal probability. The starting room and weather are also random. Your solution must discover the selected mission and current state, and adapt automatically.

1. **Cook, Cut & Clean Mission** (`cook_turkey`): cook the turkey in the oven, return it to the dining table, and cut it with the knife. Put the knife back on the table and clean the oven after it cools down. At the end, the turkey must have status `cooked_cut` and be on the dining table, the knife must also be on the table, the oven must be clean and closed, and the robot must be empty-handed.
2. **Wash & Store Mission** (`wash_jacket`): retrieve the dirty jacket from the laundry basket in room 1 and wash it in the washing machine in room 3. In sunny weather, hang it on the clothesline in room 3 until dry. In rain, use the dryer in room 1 and wait for the cycle to finish. Finally, store the jacket in the wardrobe in room 2 and close it. The jacket must have status `stored`, be inside the closed wardrobe, and have completed the weather-appropriate drying process.

Robot Services

This assignment focuses on high-level **mission planning**. The robot's actions are abstracted as ROS 2 services, and the simulator provides the state information needed by your planner. Plan and execute service calls to complete the selected mission, allowing for random initial conditions and action failures. Keep the supplied simulator, interfaces, and map unchanged; add your own ROS packages alongside them.

Service Rules

Your mission planning solution must follow these rules:

1. The robot holds at most one item. It cannot open a manual door while carrying an item, but it can operate furniture and close manual doors.
2. Opening a manual door has a 50% chance of failure. The robot must be in a connected room to operate that door. Automatic passages cannot be manually opened or closed.
3. The robot can move between rooms only through an open passage connecting them.
4. The robot must be in the same room as the furniture to interact with it. Picking and placing require open furniture; only the held item can be placed.
5. Cooking requires food inside a closed oven. Cooking makes the oven dirty and starts a ten-second cooldown. Cleaning requires a closed oven, an elapsed cooldown, and the robot in the oven's room.
6. Cutting requires cooked turkey on the open dining table and the robot holding the operational knife in the same room. Cutting changes the turkey's status to `cooked_cut`; the turkey stays on the table and the knife stays in hand.
7. Washing requires the jacket inside the closed washing machine and the robot in the same room.
8. Only washed clothes can be dried. In sunny weather, the robot must hold the jacket to hang it on the clothesline. In rain, the jacket must be inside the closed dryer, with the robot in the same room to start drying.
9. Drying takes five seconds. The jacket cannot be retrieved while drying or stored in the wardrobe before drying completes.

Submission Guidelines

Every student must implement **all four** methods, and each method must handle both missions and the weather-dependent branches:

- **State Machine** (e.g., SMACC2).
- **Behavior Tree** (e.g., BehaviorTree.CPP).
- **Symbolic Planning with PDDL** (e.g., PlanSys2).
- **LLM API**: use MCP to let the LLM call the ROS services.

Use existing libraries or frameworks for state machines, behavior trees, and PDDL planning instead of making AI build engines from scratch. Implement your own task logic using these tools.

Submit to your homework repo in a folder named HW3

Submit your code and a report:

- Provide the first three implementations as ROS 2 packages with launch files. Include runnable code for LLM/MCP and dependency and startup instructions for all methods.
- Show your system design. (e.g., a screenshot of your state machine and behaviour tree)
- Explain and compare the four designs, their ROS service integration, and how they handle random states, failures, and waiting for timed actions.
- Test each method on both missions, both washing-weather branches, and different initial rooms. Make a screenshot of the simulator completion evidence.
- Do your best for the LLM part and explain any limitations.
- Report again on your use of AI for this HW: What services did you use and what software on your computer? How much work did AI save you? Provide some example prompts.

Good luck, and happy coding!

References

- [1] Delin Feng, Chengqian Li, Yongqi Zhang, Chen Yu, and Sören Schwertfeger. Osmag: Hierarchical semantic topometric area graph maps in the osm format for mobile robotics. In *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–7. IEEE, 2023.