

Semantic Rearrangement with OVMM and a VLA Policy

Supervisor: Rohit Menon

Why this project: Build a Toyota HSR demonstration that performs RGB-D mapping, constructs a metric-semantic scene graph, plans an open-vocabulary manipulation task, and executes the local pick/place motion with a learned policy.

Learning outcome: Integrate RGB-D mapping and TF registration, scene-graph construction, task-and-motion planning, vision-language-action inference, and ROS2 control on the HSR.

Semantic Rearrangement Architecture

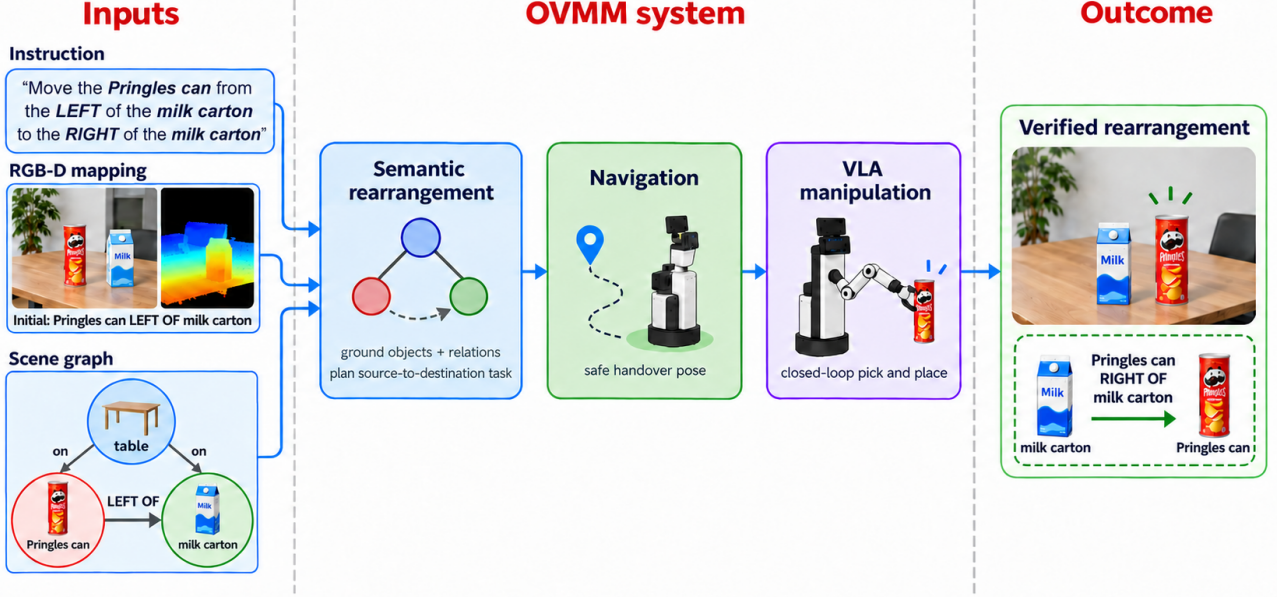


Figure 1: The scene graph records that the Pringles can is initially left of the milk carton. DORA plans the move and navigation, and the VLA places the can at a verified position right of the carton.

Motivation

DORA-OVMM provides scene-graph reasoning, navigation, and task execution, while local manipulation must close the loop on current camera and joint observations [2]. DORA handles task-and-motion planning; a vision-language-action (VLA) policy executes bounded pick/place motions. Boxer lifts open-vocabulary 2D detections into global 3D oriented bounding boxes registered in the map frame [1]. An alternative mapping method is acceptable if it provides equivalent object geometry and spatial relations. AIRoA provides an HSR-specific $\pi_{0.5}$ checkpoint and deployment code using head RGB, hand RGB, proprioception, and a language instruction [3, 4].

Goal

Implement and evaluate three additions to DORA-OVMM:

1. **Metric-semantic mapping:** construct the initial scene graph from a calibrated RGB-D mapping trajectory with synchronized TF, using Boxer or a comparable open-vocabulary 3D mapping method. The task planner receives no prior scene graph.
2. **Task-and-motion execution:** use DORA-OVMM to ground the command, select the source and destination objects, plan the rearrangement, navigate to a collision-free manipulation pose, and transfer controller ownership explicitly.
3. **ROS2 VLA control:** implement an HSR ROS2 interface to the released AIRoA inference server, execute bounded local pick/place skills, and return an independently verified result to DORA.

Available Software and Scope

Students reuse DORA-OVMM, Nav2, SAM3, the AIRoA/OpenPI code, and the released HSR checkpoint. Boxer is the recommended mapping backend; an alternative is acceptable if it outputs 3D object poses, spatial relations, and map-frame registration. Robot observations, action execution, controller arbitration, timeouts, and safety limits must run in ROS2. Model inference may run remotely through the official WebSocket server [5]. The released checkpoint initializes HSR task adaptation; zero-shot execution is evaluated rather than assumed [3].

Online scene-graph learning, foundation-model pre-training, new VLA architectures, active object search, persistent household memory, and general articulated manipulation are outside the required scope. The core demonstration uses visible tabletop objects and known destination surfaces.

System Operation

The robot executes or replays a calibrated RGB-D mapping trajectory through a previously unmapped lab or household scene. Synchronized RGB-D frames, camera calibration, and TF transforms are fused into 3D object estimates registered to the Nav2 map. These estimates are converted into room, furniture, object, containment, support, and task-relevant spatial relations. DORA then executes a command such as “bring me the snack can” or “move the red bottle to the tray.” It controls semantic rearrangement and long-range motion. At a validated handover pose, Nav2 and the classical manipulation controller stop. The VLA receives head RGB, hand RGB, eight HSR joint/gripper values, and a bounded skill instruction, and returns arm, gripper, head, and base commands. Only one controller owns each actuator at a time.

The project is designed for three students, each receiving 9 ECTS. The expected workload is approximately 270 hours per student, or 810 student-hours in total.

Work Packages

WP	Student A: Rearrangement	Student B: Scene Graph	Student C: ROS2 VLA
WP1 – Interfaces	Rearrangement goals, preconditions, handover state, and outcomes.	Rosbag format, camera calibration, TF tree, map frame, graph schema, and quality checks.	Observation/action schema, inference client, controller ownership, and limits.
WP2 – Core	Ground source/destination, plan semantic rearrangement, navigate, validate handover, and sequence skills.	Run Boxer or an alternative, fuse mapped 3D objects, estimate required relations, and export the DORA graph.	Subscribe to both RGB streams and proprioception; execute bounded $\pi_{0.5}$ action chunks through ROS2.
WP3 – Safety	Pause/resume Nav2, prevent simultaneous controllers, and recover after task failure.	Reject misregistered, inconsistent, or low-quality graphs explicitly.	Joint/workspace/rate limits, stale-input checks, timeout, zero command, and emergency stop.
WP4 – Evaluation	Grounding, planning, navigation, handover, recovery, task outcomes, and classical baseline.	3D detection, label, relative-pose, and map-registration accuracy.	Offline action inspection, local-skill success, latency, interventions, and failures.
Shared	Simulation integration, real-HSR trials, matched classical-manipulation baseline, run logging, demonstration, and final report.		

Each student should plan approximately 40 hours for setup and benchmark definition, 150 hours for the individual work package, 50 hours for integration/evaluation, and 30 hours for documentation and demonstration.

Project Schedule

Period	Planned Milestone
November	Reproduce the DORA simulator/robot baseline; define mapping-data, graph, handover, VLA, safety, and logging interfaces.
December	Build and validate one scene graph from a recorded RGB-D/TF mapping run; replay AIROA observations and inspect predicted action chunks without robot execution.
January	Integrate one simulated command from graph construction to a bounded VLA skill.
End of January	Midterm: scene graph from a new mapping trajectory, DORA semantic rearrangement/handover, and one $\pi_{0.5}$ manipulation skill in simulation or controlled replay.
February	Real-HSR ROS2 adapter, task-specific adaptation if required, and repeated single-object trials.
March	Full command-driven demonstration, matched evaluation, final presentation, and report.

Controller Handover Conditions

The VLA may start only when the requested object and destination are resolved, inputs are fresh, the robot is inside the defined manipulation region, and DORA has stopped competing motion controllers. It returns **success**, **failure**, **timeout**, or **safety-stop**. Success is established by an independent detector/contact or object-motion check, not by the policy’s own command history.

Outcome Levels

Minimum Outcome

Student A: Load the generated graph and execute one simulated DORA rearrangement command through grounding, navigation, and explicit handover. **Student B:** Convert one synchronized RGB-D/TF rosbag from a mapping trajectory into a visualized scene graph registered to the Nav2 map. **Student C:** Implement and test the ROS2 observation/action adapter using recorded inputs, inspected actions, and disabled robot output. **Integration:** Demonstrate the complete system in simulation or controlled replay: scene graph $\rightarrow$ DORA rearrangement $\rightarrow$ VLA handover.

Expected Outcome

Student A: Run repeated semantic rearrangement commands with handover checks, independent outcome verification, explicit failure reports, and a classical baseline. **Student B:** Build and evaluate graphs from multiple real-HSR mapping runs, including object, furniture, room, containment, support, and task-relevant spatial relations. **Student C:** Execute one bounded AIRoA $\pi_{0.5}$ pick/place skill on the HSR with exclusive controller ownership and safety limits. **Integration:** Perform repeated command-driven tabletop trials on the real HSR and compare VLA and classical manipulation under matched initial scenes.

Desired Outcome

Student A: Return control to DORA and recover after a missed grasp, lost target, or occupied destination. **Student B:** Update selected graph objects after execution or improve a documented graph failure case. **Student C:** Adapt the checkpoint to one selected task if zero-shot performance is insufficient, using held-out trials and documented data provenance. **Integration:** Support a second object/destination combination and demonstrate one end-to-end recovery.



Figure 2: The existing HSR setup provides navigation and manipulation; this project adds metric-semantic mapping, task execution, and VLA manipulation.

Evaluation

Failures are assigned to camera/TF calibration, RGB-D mapping and graph construction, grounding, navigation, controller handover, VLA execution, or outcome verification. Each run records the mapping method, command, graph revision, selected object and destination, active controller, observations, actions, timing, interventions, outcome, and failure reason. Classical and VLA variants use the same initial robot and object states. Zero-shot failures and other negative results remain part of the evaluation.

Risk and Contingency

Expected failure modes are map-frame registration error, poor VLA transfer, inference latency, ROS1/ROS2 interface differences, and overlapping controller commands. Initial tests therefore replay recorded RGB-D/TF rosbags with robot output disabled. The official policy server is retained, while robot I/O is implemented in ROS2. If the base checkpoint fails on the selected task, students either perform bounded task-specific adaptation or report the classical DORA executor as the baseline. If HSR access is delayed, the simulated system and offline evaluation on recorded HSR data remain required.

Prerequisites and Deliverables

Prerequisites are Python, basic ROS2, coordinate transforms, linear algebra, and experience with simulation or robot hardware. Deliverables are the RGB-D mapping and scene-graph nodes, DORA rearrangement executive, ROS2 AIRoA interface, controller handover and recovery logic, configuration files, recorded rosbags, evaluation results, setup documentation, a recorded HSR demonstration, a final presentation, and a report.

References

- [1] D. DeTone et al., “Boxer: Robust Lifting of Open-World 2D Bounding Boxes to 3D,” in *ECCV*, 2026. [Online]. Available: <https://github.com/facebookresearch/boxer>
- [2] P. Sasank, “DORA-OVMM,” software repository, 2026. [Online]. Available: <https://github.com/puresanks404/dora-ovmm>
- [3] AIRoA, “airoa-pi05-hsr-base,” model card, 2026. [Online]. Available: <https://huggingface.co/airoa-org/airoa-pi05-hsr-base>
- [4] K. Black et al., “ $\pi_{0.5}$: A Vision-Language-Action Model with Open-World Generalization,” *arXiv:2504.16054*, 2025.
- [5] AIRoA, “AiroPi for HSR: Fine-Tuning, Evaluation, and Deployment of $\pi_0/\pi_{0.5}$,” software repository, 2026. [Online]. Available: <https://github.com/airoa-org/airopi-share>