# Inha-United 2026 Team Description Paper

Minho Lee[1], Dongjin Cho[1], Minho Lee[1],
Jungtae Kim[1], Gunwoo Park[1], Gihyeok Kwon[1],
Jihyun Han[1], Sanghyun Lee[1], Wonhyuk Jung[1],
Younggun Cho[1,2], Inwook Shim[1,3], Junwoo Jang[1,4], Woojin Ahn[1,5]

[1]Inha University, Incheon, Korea,
[2]SPARO Lab.,[3]RCV Lab.,[4]Artemis Lab.,[5]RILS Lab.
Corresponding Email to:
**yg.cho@inha.ac.kr**, **iwshim@inha.ac.kr**,
**junwoo@inha.ac.kr**, **wjahn@inha.ac.kr**
`https://inha-united.github.io/Home2026/`

February 10, 2026

**Abstract.** Team Inha-United is a research-driven group formed by four robotics laboratories at Inha University. We integrate our strengths in SLAM, Large Language Model into a unified system capable of performing domestic service tasks. Using our humanoid robot RB-Y1, we aim to make the following three contributions: First, we present the intelligent robotic system that integrates a hybrid mapping approach with an LLM-based Behavior Tree. Second, addressing complex tasks, we employ dual-arm manipulation through policy learning. Third, we contribute to the RoboCup@Home community by releasing open-source resources, including dual-arm action datasets, simulated environments with robot descriptions, and modular code for manipulation and navigation tasks. The source code and datasets for our system are available at `https://github.com/inha-united-athome`.

## 1 Introduction

Our team, Inha-United, is a multi-lab research group at Inha University with diverse academic backgrounds. We have gained substantial experience across different robotic domains through a wide range of projects using various robot platforms equipped with diverse sensing modalities, as shown in Fig. 1. We have also participated in multiple international competitions listed in Table 1 as opportunities to validate our technical skills and acquire practical insights, and have received the awards indicated. Building on this foundation, we aim to realize reliable and versatile robot systems capable of performing various tasks in real-world environments and interacting naturally with humans. In this paper, we present how we integrate our modular framework into our system and outline the key components developed for the upcoming 2026 RoboCup@Home competition.


**Fig. 1.** Robot and multi-sensing platforms used in our prior research.

**Table 1.** Results of competitions

| Competitions | Results |
|---|---|
| DARPA Robotics Challenge Finals' 15 | $1^{st}$ Place |
| ICRA'24, Workshop on Construction Robots | Best Research Award |
| Virtual RobotX Challenge'19 | $1^{st}$ Place |

## 2 Approach

### 2.1 System Overview

**Hardware Setup:** To support our general-purpose home-service framework, we utilize the RB-Y1[1], a commercial mobile manipulator designed for complex indoor tasks. The platform features a height-adjustable torso and dual-arm mounted on a two-wheeled differential-drive base, offering the reachability and mobility required for indoor environments. The computing stack is dual-layered: a Jetson AGX Orin serves as the main controller, while a Jetson Thor is dedicated to perception workloads. A Livox MID-360 LiDAR is mounted on the sensor pack for spatial awareness. The two main computing units and the 3D LiDAR are time-synchronized via the Precision Time Protocol (PTP) to ensure temporal consistency across sensor data. The robot's vision system comprises two Intel RealSense D405 cameras for precise arm-end manipulation and a D435f mounted on the head. The head camera serves visual perception modules and the Vision-Language Model (VLM) for task execution.

[1] `https://www.rainbow-robotics.com/en_rby1`

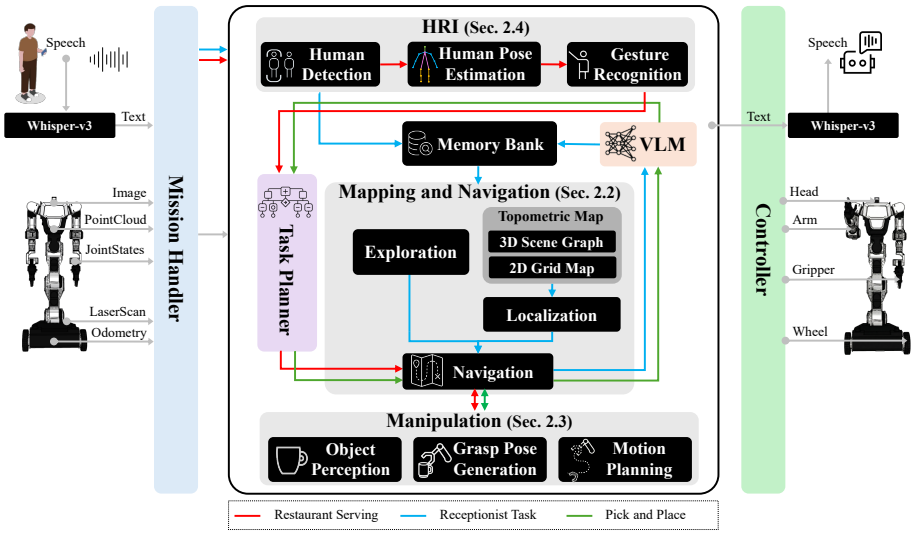


**Fig. 2.** Modular system architecture for RoboCup@Home missions, illustrating the flow from multimodal inputs through the Mission Handler to the three core capability stacks: Mapping and Navigation, Manipulation, and Human–Robot Interaction.

**Software Framework:** Fig. 2 summarizes our modular software architecture designed for three competition missions. The Mission Handler processes multimodal inputs (*e.g.*, speech, vision, and joint states), infers which mission is requested and translates it into a set of functional requirements. These requirements are dispatched to three core capability stacks: Mapping and Navigation (Section 2.2), Manipulation (Section 2.3), and Human–Robot Interaction (HRI) (Section 2.4). As shown by the colored paths (Restaurant Serving, HRI Task, and Pick and Place), each mission is executed by orchestrating reusable modules across Mapping and Navigation, Manipulation, and HRI. These modules are managed by a Behavior Tree (BT), which provides a robust framework for high-level task grounding, autonomous error recovery, and stable execution under the dynamic uncertainties of a home environment. A key feature of our system is the integration of a VLM and a Memory Bank, which allows the robot to reason about the Topometric map and store long-term environmental context.

## 2.2 Mapping and Navigation

**Topometric Map Construction:** RoboCup@Home environments often include cluttered furniture layouts and dynamic human motion, which can degrade the performance of conventional 2D SLAM. To improve robustness, our system

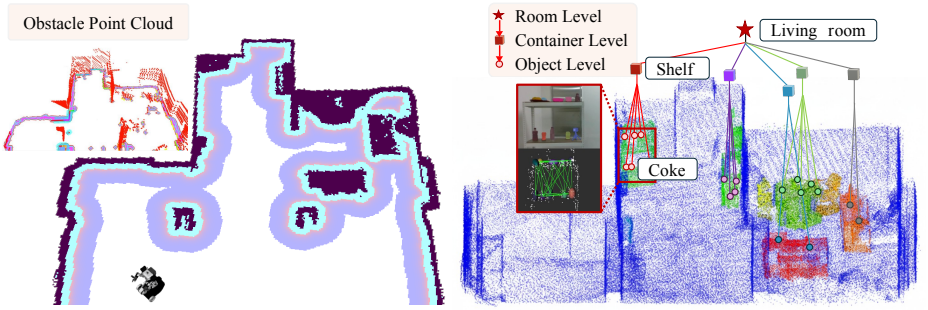


**Fig. 3.** Visualization of the hybrid map representation: 2D navigation map (left) and 3D semantic scene graph (right).

employs a hybrid 2D–3D topometric map that integrates geometric occupancy with object-level semantics. As shown in Fig. 3, our mapping module combines a 2D navigation map and a 3D semantic scene graph. The 2D navigation map is derived from LiDAR measurements by projecting an obstacle point cloud onto a 2D occupancy grid. The 3D semantic scene graph is constructed by applying open-vocabulary object detection and segmentation to RGB-D images. By maintaining semantic consistency across objects and rooms, the robot can perform object-goal navigation and scene-aware decision-making.

**Navigation:** Our navigation module leverages a 2D navigation approach[2] for the robot equipped with a 3D LiDAR sensor. Using the cost map generated in the previous stage, the system computes the global path with the A* algorithm. To handle cluttered and dynamically changing environments, we employ a DWB local planner that generates and updates the robot's trajectory based on newly received sensor data while following the global path. This allows the robot to avoid dynamic obstacles and maintain safe navigation.

**Exploration:** When navigating unmapped environments, we employ a frontier-based exploration module to guide the robot through unknown environments toward a task goal. Frontiers are defined as the boundary between explored and adjacent unexplored regions. We introduce a frontier score that prioritizes candidate frontier poses according to their expected coverage of unexplored regions, enabling efficient local navigation while incrementally building the map.

### 2.3 Manipulation

**Object Perception:** Object grasping requires acquiring target information through detection and geometric extraction. For object detection, we employ

[2] `https://www.opennav.org/`

closed-set based YOLOv8 [1] for pre-defined object list and Grounding DINO [2] for unknown objects. Then, PTv3 [3] serves as the backbone to extract 3D geometric features from the detected target. We employ multi-view-based 3D registration to mitigate noise and partial data from single-view observations. To obtain multi-view measurements without additional actions, we utilize cameras mounted on the dual-arm and head.

**Grasp Pose Generation:** We generate grasp pose from the target object modeling using diffusion-based GraspGen [4], which produces candidate grasping poses. By memorizing these candidates, our system can recover from grasp failures without repeating the manipulation process. We apply a dual-arm reachability map to improve resource efficiency by filtering out candidates beyond the reachable region. Furthermore, we utilize this map to allocate the appropriate arm by determining which arm's workspace better covers the target.

**Motion Planning:** To ensure stable object grasping while accounting for the current robot state and environmental constraints, we employ a collision map-based two-step motion planning strategy. In the first step, the end-effector is positioned at a standoff location, offset by the gripper's length from the target grasp pose along the local approach axis. Subsequently, a Cartesian path motion is applied to linearly approach the object and complete the grasp.

### 2.4 Human-Robot Interaction

**Human Perception:** The human perception module supports both identity recognition and intent inference through submodules: human detection, pose estimation, and gesture recognition. User identities are stored in a Memory Bank using InsightFace[3], allowing the robot to recognize previously encountered individuals and provide personalized interaction. Additionally, human pose information is extracted using YOLOv8-Pose [5] and interpreted as gesture cues for inferring user actions and implicit intent.

**Natural Language Understanding:** To support diverse human-robot interaction tasks involving natural language, such as interactive dialogue and instruction-following, we leverage foundation models, Whisper-v3 [6] for Automatic Speech Recognition and Gemma 3 [7] for Natural Language Understanding. Rather than maintaining separate fine-tuned models for each downstream task, the system introduces a task manager that applies task-specific prompts and routes the parsed inputs accordingly, enabling a single language model to efficiently support multiple interaction scenarios.

[3] `https://insightface.ai`

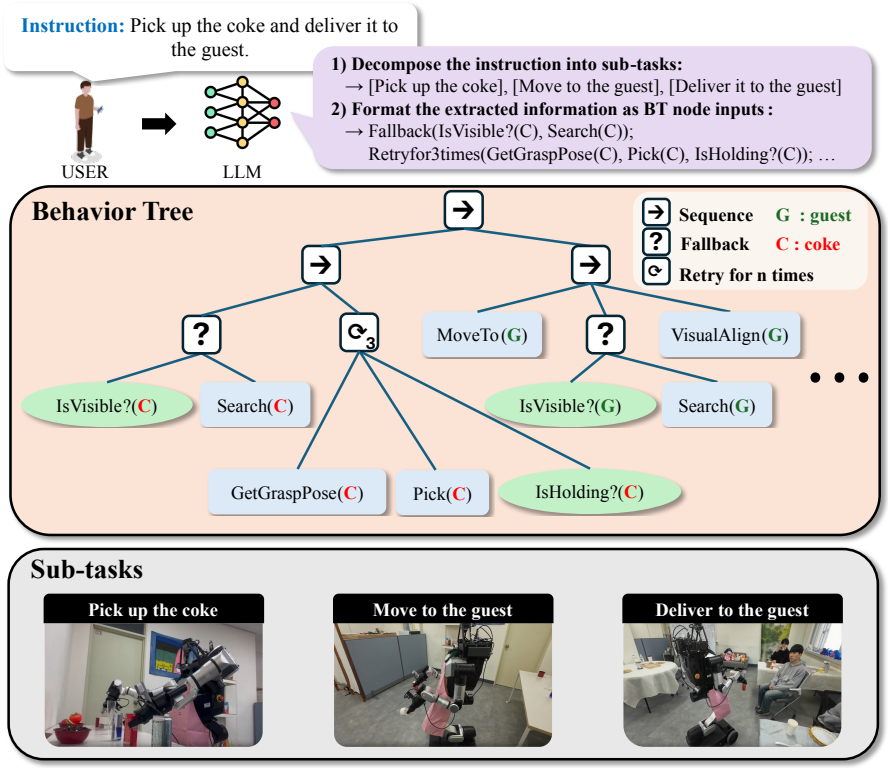


**Fig. 4.** Process of converting a natural language instruction into a BT with explicit condition checks, fallback, and retry nodes.

**Context-Aware Decision-Making:** Our decision-making module interprets user intent and environmental context to generate structured high-level commands. As shown in Fig. 4, a Large Language Model (LLM) decomposes the user instruction into subtasks and maps them to pre-built BT nodes that interface with the underlying modules. This integration leverages the stability of BT-based control and the flexibility of LLM-driven reasoning, enabling the robot to adapt to diverse tasks while maintaining predictable and context-aware behavior.

### 2.5 Simulation and Sim2Real

**Simulated Environments:** We built simulated environments in MuJoCo[4] and Isaac Sim[5], as shown in Fig. 5(a) and (b). MuJoCo enables fast controller proto-

[4] https://mujoco.org/
[5] https://developer.nvidia.com/isaac


**Fig. 5.** RB-Y1 in simulation and real-world settings, including (a) MuJoCo, (b) Isaac Sim, and (c) human demonstrations used for policy learning and Sim2Real transfer.

typing, while Isaac Sim provides high-fidelity physics for testing complex interactions. These environments allow us to evaluate navigation and manipulation behaviors under varied lighting, layouts, and occlusions before deploying to the RB-Y1. By iterating in simulation first, we can quickly identify failure cases and refine robot policies prior to real-world testing.

**Sim2Real Transfer:** The simulation environment is first constructed to closely match the real-world, and policies are initialized through training in simulation. Domain randomization is applied to both visual and physical properties, including lighting, textures, layouts, sensor noise, friction, mass, and collision parameters, to expose the policy to a wide range of operating conditions. Given the close alignment between the simulation and the real-world setup, we further apply domain adaptation using real-world demonstration data, with Fig. 5(c) illustrating the data collection process. This adaptation is performed through two complementary strategies: (i) learning domain-invariant feature representations, and (ii) policy-level fine-tuning using real-world demonstrations that provide robust feedback to the simulation-trained policy. Specifically, the policy is trained to prioritize task-relevant geometric cues over low-level pixel variations, enabling stable generalization across both simulated and real environments.

## 3 Contribution

In this section, we present our three main contributions. **First**, we present an intelligent robotic system that integrates a hybrid mapping approach with LLM-based BT. By interpreting natural language instructions and grounding them into spatially structured maps through the BT, the system supports generalized task execution across diverse environments. **Second**, complex tasks such as laundry folding require multi-contacts to be maintained simultaneously, which is hard to achieve through sequential single-arm manipulation. Therefore, we employ dual-arm manipulation through policy learning to enable handling of such tasks. **Third**, we contribute to the RoboCup@Home community

by releasing open-source resources, including dual-arm action datasets, simulated environments with robot descriptions, and modular code for manipulation and navigation tasks. These resources are available at: `https://github.com/inha-united-athome`.

## 4 Conclusion

In this paper, we present the overall design of our home-service robot system, including the RB-Y1 platform. We introduced three key contributions: an intelligent task execution framework integrating LLM-based BT with hybrid mapping, dual-arm manipulation capabilities for complex domestic tasks, and open-source resources including action datasets and simulated environments. We aim to achieve full-mission execution through quantitative evaluation of each methodology across scenarios. Moreover, we will extend our system from dual-arm manipulation to whole-body planning with torso control for expanded workspace coverage. We expect to demonstrate robust and flexible performance at the 2026 RoboCup@Home competition.

## References

1. Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
2. Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.
3. Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024.
4. Adithyavairavan Murali, Balakumar Sundaralingam, Yu-Wei Chao, Wentao Yuan, Jun Yamada, Mark Carlson, Fabio Ramos, Stan Birchfield, Dieter Fox, and Clemens Eppner. Graspgen: A diffusion-based framework for 6-dof grasping with on-generator training. *arXiv preprint arXiv:2507.13097*, 2025.
5. Debapriya Maji, Soyeb Nagori, Manu Mathew, and Deepak Poddar. Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2637–2646, 2022.
6. Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.
7. Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

## Inha-United@Home RB-Y1 Robot Hardware Description

Specifications of the RB-Y1 robot (Fig. 6):

- **Base:** 2-wheel differential-drive with rear caster (1.5 m/s max speed)
- **Dimensions:** height: 1.68 m (max), width: 0.6 m, depth: 0.69 m
- **Weight:** 131 kg
- **Head/Torso:** 2-DoF, 6-DoF
- **Dual-arms:** 7-DoF, arm reach: 0.75 m
- **Battery:** 50 V, 30 Ah (1,500 Wh)
- **Computing Units:**
  - **Main PC:** NVIDIA Jetson AGX Orin 64 GB
  - **Perception PC:** NVIDIA Jetson Thor 128 GB
  - **Internal Control PC:** UP Xtreme i12 with 12th Gen Intel® Core™ processor

**Fig. 6.** Inha RB-Y1

*Also our robot incorporates the following sensors:*

- **LakiBeam 2D LiDAR** used for 2D SLAM and navigation
- **Livox MID-360 3D LiDAR** for omnidirectional perception
- **Intel RealSense D435f** with head mount and **Intel RealSense D405** with wrist mount
- **ReSpeaker Mic Array v3.0** used for voice acquisition

## Robot's Software Description

*For our robot we are using the following software:*

- **2D Mapping:** SLAM ToolBox
- **Motion Planning:** MoveIt2
- **Grasp Pose Generation:** GraspGen
- **Automatic Speech Recognition:** Whisper-v3
- **Large Language Model:** Google Gemma3
- **Object Recognition:** Grounded-SAM2, YOLOv8
- **Human Pose Detection:** YOLOv8-Pose
- **Face Recognition:** InsightFace

## External devices

*Our robot relies on the following remote high-performance computing resource:*

- **Remote High-Performance Computing Server:** Two NVIDIA RTX A6000 GPUs, 256 GB RAM, 8TB SSD
- **Desktop:** Intel Core Ultra 9 285K CPU, 64 GB RAM, 1 TB SSD, NVIDIA RTX 5090 GPU