RoBorregos Team Description Paper for RoboCup@Home 2026

José Domínguez Alejandro González-Escamilla
Gilberto Malagamba Danaé Sánchez Paola Llamas
Gerardo Fregoso Jocelyn Velarde Emil Winkler Camila Tite
Fernando Hernandez-Cantu Daniel Hinojosa Luis Benvenuto
Samuel Granados Jesus Guerrero Erick Ramírez-Cedillo
Alberto Muñoz Carlos Vázquez

November 30, 2025

School of Engineering and Sciences, Tecnológico de Monterrey
C.P. 64849, Monterrey, N.L., México

https://athome.roborregos.com/
https://github.com/RoBorregos/home2

Abstract. This paper introduces the architecture and development of FRIDA (Friendly Robotic Interactive Domestic Assistant), focusing on a hybrid neuro-symbolic approach for General Purpose Service Robots (GPSR) task. We present three main contributions: (1) A novel heatmap-based pose generation algorithm for clutter-aware placing, achieving a 95% success rate; (2) A fine-tuned LLM-driven command interpreter which parses natural language instructions into grounded and structured command lists, an approach which proves to succeed 96.5% of attempts deployed on the robot's hardware; and (3) A custom fractal-structure soft gripper designed for adaptive grasping of household objects. These systems are integrated via ROS 2 on a multi-layered software architecture and operate on task managers which follow the Finite State Machine paradigm to address the dynamic challenges of the RoboCup@Home 2026 OPL.

1 Introduction

RoBorregos is the representative team of Tecnológico de Monterrey dedicated to participating in autonomous robotics competitions since 2015. The team is composed exclusively of undergraduate students from various academic disciplines. Development for the @Home league began in 2019, debuting in Mexico's

@Home Beginners category in 2021 and advancing to Open Platform League in 2023 with FRIDA (Friendly Robotic Interactive Domestic Assistant). The team rapidly entered the international stage, achieving 16th place at RoboCup 2024 and improving to 6th place the following year.

In addition, RoBorregos has strengthened its participation in academic research through collaboration with ITESM's school of engineering. Paper outlines the team's ongoing research and development efforts towards a robust system capable of consistently executing the tasks of RoboCup@Home competition.

2 Research focus and interests

The core philosophy of the FRIDA project is focused on bridging the gap between Semantic Reasoning and Reliable Actuation in unstructured domestic environments. Driven by this goal, our research for the 2026 cycle targets specific bottlenecks in service robotics to provide reproducible contributions to the community. This year, we report significant progress in:

1. Context-Aware Manipulation: Fused environmental heatmaps with task constraints to generate optimal placing poses in clutter, addressing the "last-centimeter" problem.
2. Latency-Optimized Perception: Implemented hierarchical parallel filtering to prioritize near-field detail, significantly reducing planning latency without stalling the loop.
3. Reliable Neuro-Symbolic Grounding: Developed a synthetic dataset pipeline to fine-tune LLMs for reliable instruction grounding, a contribution recently published in MICAI. [1]
4. Segmentation Dataset Pipeline: Upgraded data generation from bounding boxes to pixel-level masks, yielding cleaner annotations and robust training data.
5. Zero-Shot Object Detection: Deployed a YOLO-E detector to recognize objects via text prompts alone, enabling rapid integration without retraining.

3 Robot architecture

FRIDA operates on a distributed architecture built using ROS 2 Humble Hawksbill, leveraging the Data Distribution Service (DDS) for real-time, fault-tolerant communication between modules. The system is designed around a centralized Task Orchestrator that coordinates asynchronous independent sub-systems.

3.1 Containerized Deployment

The software stack is containerized using Docker, which enables multi-platform deployment with automated environment detection and isolation of complex dependencies for the sub-systems.

3.2 Modular Design and Standardized Communication

The system is partitioned into four modules with their own set of ROS nodes: Vision, HRI, Manipulation, and Navigation. By leveraging standardized ROS 2 interfaces, the underlying logic of each area is abstracted, allowing for a clear separation of concerns and a microservice-style architecture.

4 Scientific contributions and current work

This section delves into further detail on the highlights of development and achievements per area.

4.1 Integration

Implemented framework using modern software engineering practices: abstracting hardware functionalities, enforcing module isolation, relying on a structured source control with containerized CI/CD pipelines, unified in a multi-layered system.

Software Architecture. The operational flow begins with an input signaling the start of the task; next a task-specific deterministic Finite State Machine (FSM) orchestrates the high-level logic; here sub-task coordinators provide abstracted functions for calling nodes at module-level; at the lowest level, modules interact with the hardware and execute primitive actions.

Task Planning. The use of FSMs facilitates logic-driven translation of complex tasks into concrete and robust robotic behaviors. It allows the incorporation of dynamic planning and fault tolerance which enables: receiving feedback for grounding the robot's actions, ease of testing by mocking responses from certain modules, and the incorporation of retry mechanisms in case sub-task functions fail, along with fallback policies involving asking humans for help (Deus ex machina).

4.2 Object Manipulation

The developed pipeline enables interaction with a 6-degree-of-freedom (DOF) robotic arm and its gripper to perform manipulation tasks specifically picking and placing objects, as well as pouring cereal into a container by integrating image processing, perception, and motion planning components.

Motion Planning Strategy. The system employs a hybrid architecture where Sampling-based Planning (RRTConnect) [2] generates collision-free trajectories for manipulation, enforcing strict orientation constraints during pouring to ensure stability. Conversely, auxiliary tasks (homing, scanning) utilize direct joint interpolation for high-speed movement in free space.

Picking Task. To accurately isolate target object clusters, 2D detection data is fused with 3D point clouds. To minimize latency, the pipeline applies parallel filtering and variable-resolution downsampling. The planning scene is optimized by modeling the table as a primitive box and obstacles as spheres via RANSAC [3]. Finally, GPD [4] interprets object clusters, enabling MoveIt! [5] to generate collision-free trajectories within this simplified environment.

Placing Task. The table surface, isolated via RANSAC [3], generates a Gaussian base heatmap favoring accessible space. This is combined with a "Special Request Heatmap" (encoding relative constraints like "in front of") to determine the optimal placing pose from the combined costmap peak [Fig. 2].

Pouring Strategy. The container's 3D cluster is analyzed via PCL [6] to determine its centroid and rim height. A `pour_pose` is defined above the center, retaining the gripper's pick orientation. The `PourMotion` server uses MoveIt! [5] with dynamic orientation path constraints to prevent spillage.

Custom Gripper. A custom parallel gripper, powered by a rack-and-pinion mechanism and high-torque servo, secures objects up to 1 kg. Its novelty lies in fingers incorporating fractal-structured silicone covers [7] and soft inner materials, enhancing deformation and adaptability to complex geometries [Fig. 4].

Experimental Validation. Robustness was evaluated through 40 trials (20 picking, 20 placing), with randomized configurations of household objects (e.g., an apple) to test adaptability [Fig. 3]. The results are shown in Table 1.

Picking Reliability. The sequential protocol with randomized clutter achieved an 80% success rate. Failures stemmed primarily from motion planning constraints in high density scenarios and dynamic interactions with spherical objects (e.g., orange rolling upon contact).

Placing Reliability. The pipeline demonstrated high robustness with a 95% success rate. Heatmap logic effectively directed placement to free space. The single failure resulted from a minor collision with an adjacent object during retraction, though the placed object remained stable.

Table 1. Manipulation Success Rates (20 Trials)

Task Phase	**Successes**	**Attempts**	**Success Rate (%)**
Object Picking	16	20	80%
Object Placing	19	20	95%

4.3 Computer Vision

Face recognition. DLib's pretrained ResNet-based face-recognition model [8] was used for fast, high-accuracy identification, generating 128-D facial embeddings for each detected face. Encodings are computed once per detection and reused for subsequent recognition. To reduce computational cost, detections are associated across frames using a spatial tracking heuristic.

Person tracking and re-identification. YOLO11 with ByteTracker [9] provides short-term identities, while a Re-ID module restores IDs after occlusion or re-entry. Multiple models were trained on Market1501 using the Person reID baseline [10], with Swin selected for its stronger embeddings. Matching is performed via cosine similarity.

To maintain real-time performance, Re-ID runs sparingly; embeddings are computed once and reused with multi-view additions for robustness. ByteTracker narrows candidate matches, and inference is GPU-optimized to sustain camera-rate processing.

Pose and gesture recognition. MediaPipe [11] extracts skeletal landmarks used for gesture inference via rules on joint height, limb orientation, and inter-joint angles (e.g., an arm is raised when the elbow remains above the shoulder). To complement this, a vision–language model (VLM) is integrated for higher-level interpretation. The Moondream 2 model [12] with SigLIP and Phi-1.5 weights, processes cropped person images and returns natural-language descriptions, including pose and clothing attributes.

Zero-shot object detection. A YOLOE based zero-shot detector was implemented to identify objects via vision–language embeddings rather than fixed classes. This enables prompt-based detection but yields weaker performance on common items compared to supervised models [Fig. 5]. The system supports RGB–depth processing and publishes 2D detections, 3D poses, RViz markers, and annotated images with dynamic class configuration.

Object detection. A dataset pipeline was developed using GroundingDINO [13] for zero-shot detection, which finds objects from user-provided class names, and SAM [14] for mask refinement, producing clean transparent-object crops with minimal manual intervention.

These isolated assets are then composited onto real backgrounds to generate synthetic samples with automatic labels and augmentation, yielding training-ready datasets for object detection and segmentation. Additionally, the YOLO model was exported to TensorRT, reducing memory usage by 50% and nearly doubling inference speed, significantly improving deployment efficiency. Across 20 images containing 100 annotated objects, the detector achieved 86 correct identifications, corresponding to an overall success rate of 86%.

4.4 Human-Robot Interaction (HRI)

Speech. The main communication interface with users is through speech. Speech To Text is performed via an adaptation of the WhisperLive [15] implementation using a faster-whisper backend. This enables interpreting a stream of output text and providing feedback or taking action whilst the user is speaking. TTS is also performed live, leveraging the open-weight, fast, and efficient model Kokoro-82M [16]. Together, they provide a seamless and time-sensitive user experience. These modules are containerized outside the ROS environment by leveraging lightweight and high-performance gRPC communication.

Display. A second user interface is provided in the form of a display in the robot's base. It provides insights into FRIDA's algorithms, allowing users to better understand how and when to interact. Some examples include: icon prompting users to speak, live STT transcriptions, TTS dialog, detected keywords, vision detections with bounding boxes, and input audio levels. The web UI is implemented in Next.js and connected via WebSockets to a Node.js backend, which forwards ROS2 messages using the rclnodejs client library for JavaScript.

Command Interpreter. This pipeline translates natural language into robot-executable tasks. The process is handled by a locally hosted fine-tuned Llama 3.1 8B model, served via the Ollama framework, trained on a synthetically generated dataset.

The dataset is generated through a structured pipeline that begins with a set of predefined command templates. These templates are automatically populated with variable entities (e.g., names, locations, and objects), after which additional follow-up instructions are randomly selected and appended. Each entry in the dataset comprises two synchronized outputs: a paraphrased natural-language instruction, and a corresponding structured action plan composed of predefined primitive actions.

To ensure robust command generation, the BAML framework is integrated as a proactive validation layer. By defining a strict data schema, the LLM's output is forced into a correct, parsable JSON format, avoiding action-level hallucinations. With this research RoBorregos published a paper winning 2nd Best Student Paper at the 24th Mexican International Conference on Artificial Intelligence [1].

Retrieval-Augmented Generation for Grounding. To guarantee that all commands are grounded in the robot's real world knowledge, they are processed by a RAG system. When a user's instruction is transcribed, the NLP service first queries the persistent PostgreSQL Knowledge Base. This retrieval step uses the pgvector extension to perform an efficient similarity search. It fetches relevant context such as the known location of an item, properties of an object, or details about a person, which is then dynamically injected into the command interpreter prompt. With this retrieved context, the LLM is asked to generate a command

that is not only syntactically correct but also factually grounded in the robot's context.

4.5 Navigation

FRIDA's navigation stack builds on RTABMap for SLAM and Nav2 for path planning and control. The robot currently uses a differential-drive Dashgo B1 base, while an omnidirectional platform is under active development for future iterations.

Navigation stack overview. The navigation system relies on RTABMap to generate a 3D SLAM map and a 2D occupancy grid used for global planning. Global paths are computed over this costmap using an A* based planner within the ROS 2 Nav2 framework, while a Model Predictive Path Integral (MPPI) controller tracks the planned trajectory and generates smooth, dynamically feasible velocity commands [17].

Base and state estimation. FRIDA employs a modified EAI Dashgo B1 differential-drive chassis, interfaced through an STM32 based controller with a custom ROS 2 driver. Odometry is obtained by fusing wheel encoder data and IMU measurements via an Extended Kalman Filter (EKF), improving short-term pose accuracy and robustness to wheel slip.

Behavior tree and costmaps. Task-level navigation is structured as a behavior tree using Nav2's BT-based architecture, which coordinates goal reception, planning, recovery behaviors, and path execution. Local and global costmaps integrate data from the RP Lidar A1 and RGB-D perception, enabling obstacle inflation and dynamic obstacle avoidance while selecting efficient and safe paths.

SLAM and sensor fusion. RTAB-Map operates in visual-SLAM mode using RGB-D input, and can additionally incorporate LiDAR data to improve loop closure and map consistency. Visual features are extracted with an ORB-based front end, and the resulting pose estimates are aligned with the EKF odometry frame to provide a consistent navigation reference for both the differential base and the planned omnidirectional platform.

5 Conclusions and future work

The contributions to the @Home league include general-purpose modules for each of the development areas, integrated into a cohesive layered architecture. Participation in RoboCup 2025 provided valuable insights into FRIDA's system maturity under competitive conditions. Moving forward, the team is committed to enhancing performance in real-world tasks, spanning from HRI to low-level

control systems. A key priority is the migration to an omnidirectional base to improve the mobility and maneuverability of the robot. We are currently working on the development of the omnidirectional base and contributing improvements to the open-source ecosystem that supports it.

References

1. Adán Flores-Ramírez et al. Taming the llm: Reliable task planning for robotics using parsing and grounding. In Lourdes Martínez-Villaseñor, Roberto A. Vázquez, and Gilberto Ochoa-Ruiz, editors, *Advances in Soft Computing*, pages 296–312, Cham, 2026. Springer Nature Switzerland.
2. James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. IEEE International Conference on Robotics and Automation*, volume 2, pages 995–1001. IEEE, 2000.
3. José María Martínez-Otzeta, Itsaso Rodríguez-Moreno, Iñigo Mendialdua, and Basilio Sierra. Ransac for robotic applications: A survey. *Sensors*, 23(1):327, 2022.
4. Andreas Ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.
5. Sachin Chitta, Ioan Sucan, and Steve Cousins. Moveit![ros topics]. *IEEE robotics & automation magazine*, 19(1):18–19, 2012.
6. Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4. IEEE, 2011.
7. Mario Martínez, Enrique Cuan-Urquizo, Erick Ramírez-Cedillo, and Armando Roman-Flores. Out-of-plane mechanical properties of additively manufactured fractal reinforced structures. In *Solid Freeform Fabrication 2023: Proceedings of the 34th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference*, 08 2023.
8. PyPI. face-recognition 1.3.0. `https://pypi.org/project/face-recognition/`, 2020.
9. Yifu Zhang et al. Bytetrack: Multi-object tracking by associating every detection box. In *European Conference on Computer Vision*, pages 1–21. Springer, 2022.
10. Zhedong Zheng et al. Joint discriminative and generative learning for person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2138–2147, 2019.
11. Camillo Lugaresi et al. Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*, 2019.
12. Moondream. Moondream ai. `https://moondream.ai/`, 2024.
13. Shilong Liu et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2024.
14. Nikhila Ravi et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.
15. Collabora. Whisperlive. `https://github.com/collabora/WhisperLive`, 2025.
16. Hexgrad. Kokoro-82m (revision d8b4fc7), 2025.
17. Sagarnil Das. Simultaneous localization and mapping (slam) using rtab-map. 2017. SLAM Project, Robotics Nanodegree Program, Udacity.

FRIDA Hardware Description

- Base: A modified EAI Dashgo B1 differential drive robot was used for mobility purposes and motors with encoders for odometry purposes.
- Computers: Jetson Orin AGX.
- Batteries: A total of three 24v li-ion batteries are used, one for the xArm, Jetson Orin and Dashgo.
- Manipulator: xArm 6.
- Gripper: 2 finger parallel custom 3D printed gripper with force resistive sensors on each finger and an IR sensor for detecting objects in front of it. It uses an ESP32 as the microcontroller.
- Vision: Stereolabs ZED2 camera attached to the EOF of the xArm that runs the vision tasks and transmits data through a USB cable to the Jetson Orin.
- HRI: Seed Studio ReSpeaker Microphone array, Logitech X100 Wireless Portable Speaker, and a HDMI LCD touch screen 7" display.
- Navigation: RP Lidar A1.
- Base cover: Custom acrylic manufactured structure to protect and secure the interior electronics and components.
- Robot dimensions: Height 180 cm (fully extended), base footprint 50 cm × 42 cm
- Robot weight: 40 kg.

Also the robot incorporates the following devices:

- Power indicator for the xArm 6 and Jetson Orin batteries.
- 5V switch that enables local data transmission through an ethernet network.
- Independent emergency stop devices for xArm controller and Dashgo.

Fig. 1. FRIDA

FRIDA Software Description

- Platform: ROS 2, Humble distribution.
- Integration: Docker
- Navigation: ROS 2 Navigation Stack, RTAB-Map
- Person analysis: Torch, DLib's face-recognition, Mediapipe, Bytetrack, Moondream
- Speech recognition: WhisperLive with faster-whisper
- Keyword spotting: Open Wake Word
- Text processing: Llama 3.2, sentence-transformers/ all-MiniLM-L6-v2
- Text To Speech: Kokoro-82M
- Object detection: GroundingDINO, YOLO11, Segment anything, YOLOE
- Manipulation planning: MoveIt!
- Point cloud processing: PCL, RANSAC

Annex

Appendix 1

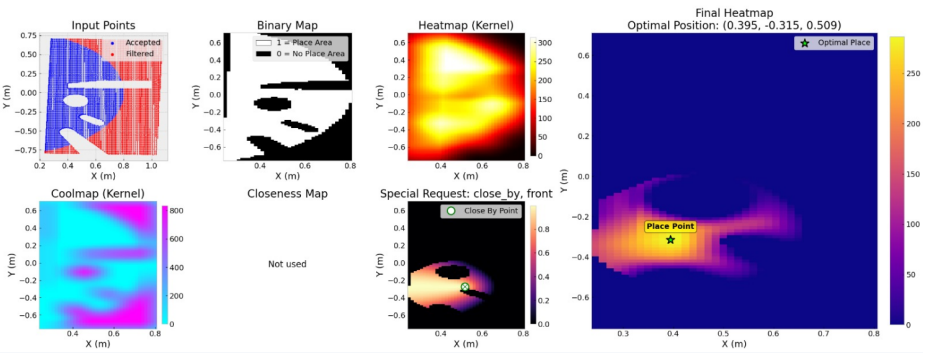

Fig. 2. Heatmaps generated to attend a request to place an object in front of another one.

Appendix 2

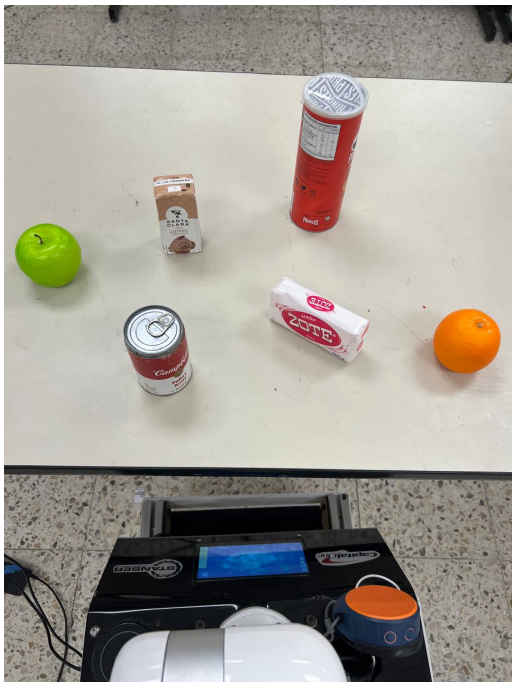

Fig. 3. Experimental setup for Pick & Place tasks.

Appendix 3

Fig. 4. Gripper grasps from different positions.

Appendix 4

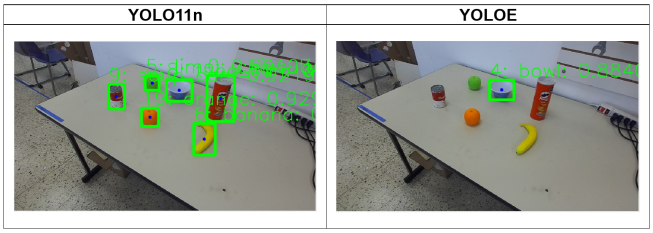

Fig. 5. Comparison between YOLO11 (supervised) and YOLO-E (zero-shot) outputs. YOLO-E detects only the bowl, its default class, while other objects require explicit text-based class definitions.