Simposio CEA de Robótica, Bioingeniería, Visión por Computador y Automática Marina 2026

RBVM
Simposio CEA de Robótica, Bioingeniería, Visión por Computador y Automática Marina

Local multi-agent LLM architecture for AAL and elderly social robotics

Warnants, I.[a,*], Roudil, F.[b], Ortiz, F.J.[a], Garrigós F.J.[c]

[a] Dpto. Ingeniería Eléctrica, Automática Industrial y Tecnología Electrónica, Universidad Politécnica de Cartagena member of European University of Technology EUT+, Campus Muralla del Mar, s/n, 30202 Cartagena, España.
[b] CPE Lyon, Graduate School of Engineering, Electrical and Computer Engineering Department, Robotics Specialization, Lyon, France.
[c] Departamento de Electrónica, Tecnología de Computadoras y Proyectos, Universidad Politécnica de Cartagena member of European University of Technology EUT+, Plaza del Hospital, 1, Cartagena 30202, Spain

Resumen

Este artículo presenta una arquitectura multiagente local orquestada mediante un modelo de lenguaje (LLM), diseñada para aplicaciones de Vida Asistida Ambiental (AAL) con robots de asistencia dentro del proyecto ADDIM. El sistema integra un robot social y dispositivos domésticos inteligentes para apoyar la autonomía de las personas mayores mediante interacción en lenguaje natural. A diferencia de los enfoques basados en la nube, todo el razonamiento de alto nivel se ejecuta localmente en una unidad de computación en el borde, garantizando preservación de la privacidad, baja latencia y funcionamiento robusto ante fallos de conectividad. Se propone el Framework ADDIM-LEAF, una arquitectura modular que incorpora un Sistema Multiagente especializado encargado de orquestar la robótica, el control del entorno, la interacción con el usuario y la monitorización del bienestar. Los agentes son coordinados por un supervisor central que emplea encadenamiento de indicaciones para la descomposición estructurada de tareas, incorporando mecanismos de validación y confirmación explícita para asegurar la seguridad operativa.

Palabras clave: Arquitectura de control para sistemas multiagente, LLMs para modelado y control, interacción humano-robot, interfaces de lenguaje natural, robótica impulsada por IA, Robótica social y ética.

Abstract

This paper presents a local, LLM-orchestrated Multi-Agent architecture designed for Ambient Assisted Living (AAL) with assistive robots within the ADDIM project. The system integrates a social robot and smart-home devices to support elderly autonomy through natural-language interaction. Unlike cloud-dependent approaches, all high-level reasoning is executed locally on an edge computing unit to ensure privacy preservation, reduced latency, and robust operation during connectivity failures. We propose the Local Edge-based Assistive Framework (ADDIM-LEAF), a modular architecture that integrates a specialized Multi-Agent System (MAS) responsible for orchestrating robotics, environment control, user interaction, and well-being monitoring. These agents are coordinated by a central supervisor using prompt chaining for structured task decomposition. The architecture prioritizes operational safety by incorporating explicit confirmation loops and validation mechanisms to constrain LLM outputs within certified boundaries. Experimental validation in a real domestic setup demonstrates the system's feasibility, responsiveness, and reliability in handling complex, multi-intent requests.

Keywords: Control architecture for Multi-Agent Systems, LLMs for modelling and control, Human-robot interaction, Natural-language interfaces, AI-powered robotics, Social robotics and ethics.

1. Introduction

The increasing pressure on healthcare systems due to Europe's ageing population, combined with the high cost of full-time homecare services, has created an urgent need for intelligent environments that support elderly autonomy safely and affordably. To address this, the ADDIM project (ADDIM Project, 2025) provides a domestic technological platform that integrates ambient sensors, wearable devices, and a socially assistive robot (SAR) to monitor well-being and facilitate natural interaction. While the initial system successfully employed Behaviour Trees for structured control, this approach proved limited by its rigidity, requiring every interaction to be explic-

*Autor para correspondencia: ismael.warnants@edu.upct.es

itly programmed in advance, which is both time-consuming and difficult to scale as user requests become more varied.

To overcome these limitations, we propose a Multi-Agent System (MAS) architecture that leverages a Large Language Model (LLM) to provide a more flexible and adaptive coordination layer. This paper focuses on the design and integration of a Multi-Agent System into ADDIM-LEAF (Local Edge-based Assistive Framework), a unified decision-making architecture that interprets natural-language requests and orchestrates tasks across specialized agents. The framework utilizes prompt chaining for structured task decomposition and consistent instruction following to provide a seamless assistive experience. Unlike typical cloud-based AI, our architecture is deployed entirely on a local edge computing unit (NUC) and utilizes a Mosquitto (MQTT) server for inter-component communication. This ensures that the system remains reliable during connectivity failures and preserves user privacy by processing all data within the home.

The main contribution presented in this paper is a safety-oriented local orchestration layer that integrates generative reasoning with deterministic safeguards. To ensure stable operation in sensitive elderly-care environments, the system incorporates explicit confirmation loops, requiring user validation before executing sensitive actions. The proposed system is currently undergoing real-world validation in two distinct settings: a home of an elderly woman in Cartagena and an adapted apartment at the elderly care centre of Fundación Poncemar in Lorca (Murcia).

2. Related work

Technological development in Ambient Assisted Living (AAL) and domestic robotics has historically been fragmented. Early systems often suffered from high operational complexity and a heavy reliance on cloud connectivity. This dependence is particularly risky in elderly care, as it introduces critical failure points during emergencies and raises serious privacy concerns regarding sensitive data (Anghel et al., 2020).

While commercial smart speakers like Amazon Alexa or Google Home provide accessible voice interfaces and improve social engagement, they lack mobility and are limited by always-listening cloud-based models that lack the decision-making autonomy required for dedicated AAL architectures (Dino et al., 2025; Kim and Choudhury, 2021).

Traditional AAL control relies on deterministic methods like rule-based engines or Behaviour Trees (BTs) to ensure predictable robot behaviour (Iovino et al., 2022). Although BTs are modular and transparent, they are often too rigid for domestic use because every potential interaction must be hard-coded. This makes it difficult for systems to handle the ambiguous nature of natural human speech.

To improve scalability, researchers introduced Multi-Agent Systems (MAS) to distribute tasks across specialized entities (Blake et al., 2025). More recently, Large Language Models (LLMs) have enabled more natural dialogue and flexible task planning. For example, LLMs have been used on NAO robots to enhance conversation (Casasola et al., 2025) and to automatically generate robot Behaviour Trees from natural-language (Fidalgo et al., 2025). Recent improvements, such as the Failure Interpreter module, allow these systems to adapt plans dynamically when tasks fail, reaching high success rates in home testing (Merino-Fidalgo et al., 2025).

Despite these advancements, most current LLM-based systems still depend on cloud services. This is a significant drawback for elderly care, where privacy and constant reliability are non-negotiable. This paper addresses these issues by moving LLM coordination entirely onto a local edge unit. By processing reasoning in-house and separating it from physical control, we provide a flexible system that preserves user privacy and maintains functionality without an internet connection.

3. ADDIM System Architecture

The ADDIM system is a hardware-software platform designed to monitor the emotional and functional state of elderly individuals through a non-invasive approach. As illustrated in Figure 1, the architecture is organized into three distinct layers that manage data from the home environment to the final clinical analysis, all hosted within an Intel NUC edge device (Intel NUC 11 Enthusiast NUC11PHKI7CAA0 GeForce RTX 2060 6GB or similar) inside the home of the user to ensure privacy and security.

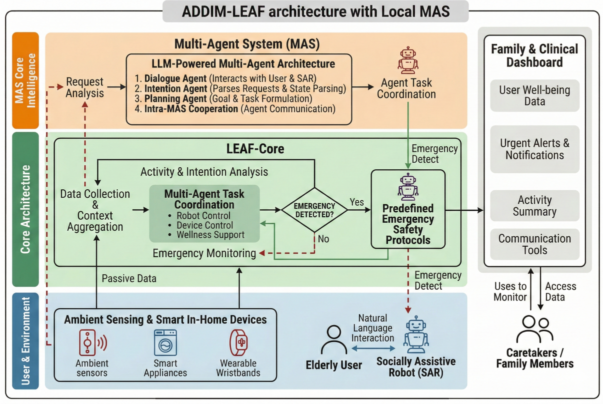

Figure 1: The ADDIM-LEAF architecture with a Local MAS.

The bottom layer utilizes a Social Robot (OrionStar Mini-bot) equipped with a custom Android application as the primary interface. The user is equipped with a Pixel Watch 3 to monitor physiological data, while the environment is monitored and controlled by a Zigbee sensor network. This network includes pressure sensors in seats and beds, contact sensors for doors and drawers, power consumption plugs for appliances, infrared motion detectors in every room, smart lights, smart curtains, and wireless emergency buttons. The local network is managed by a TP-Link Wi-Fi router and an Uninterruptible Power Supply (UPS) is used to support both the router and the NUC edge computing unit to ensure 24/7 reliability and data integrity during power or connectivity fluctuations.

The orchestration layer is comprised of the ADDIM-LEAF (Local Edge-based Assistive Framework) software architecture programmed in Python that integrates the LEAF-Core with the Multi-Agent System (MAS). It manages data collection, context aggregation, and intention analysis from sensors and wearables, utilizing the LEAF-Core for monitoring and the integrated MAS for high-level intelligence and complex task handling. This system allows for continuous passive monitoring, providing round-the-clock observation that acts as a supportive tool for caregivers. This role allows elderly individuals to live more independently and securely, knowing they will receive help if an emergency is detected.

Replacing the rigid Behaviour Trees (BTs) of previous versions, ADDIM-LEAF introduces a dual-path execution logic. The LEAF-Core maintains safety via deterministic BTs for emergency protocols while collaborating with the integrated MAS to process complex, non-emergency requests. This hybrid approach enables the system to manage varied natural-language tasks that were impossible with static programming while preserving high-reliability safety standards.

The orchestration layer also acts as the bridge to clinical and support services. Caregivers and family members use this interface to monitor well-being data, receive urgent alerts, and view activity summaries. While the architecture supports remote virtual visits and robot teleoperation, it remains a decision-support tool rather than a diagnostic one.

Within this framework, the Multi-Agent System (MAS) now acts as the central coordinator. It performs request analysis and utilizes a specialized LLM-powered architecture consisting of Dialogue, Intention, Planning, and Intra-MAS Cooperation agents. This modular design allows for future addition of capabilities like computer vision or emotion detection. All core reasoning is performed locally with Ollama on the NUC using an offline Llama 3.1 8B model, ensuring the system remains functional even without an internet connection.

4. Multi-agent Supervisor Architecture

To overcome the limitations of Behaviour Trees, we integrated a Multi-Agent System (MAS) as the high-level decision-making layer within ADDIM-LEAF. Developed in Python using LangChain, this system employs a Primary Assistant to orchestrate and delegate tasks to specialized agents while maintaining a system state decoupled from the LLM. By keeping the ground truth of robot location and device status separate from the generative process, the Primary Assistant prevents hallucinations and ensures decisions are based on the actual environment state.

4.1. Specialized Agents and Their Roles

The system is organized into four sub-agents under the Primary Assistant, as shown in the MAS graph in Figure 2. Each agent is implemented as a LangChain agent with access to specific tools for their domain.

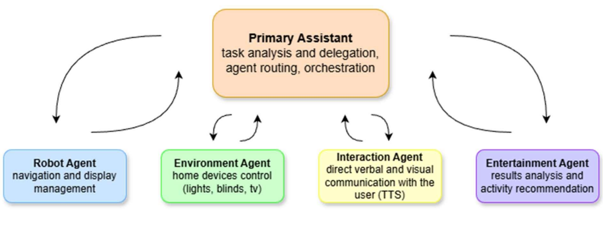

Figure 2: MAS graph

The Robot Agent is in charge of navigation and the robot interface. The Environment Agent controls home hardware like lights and blinds by using specific tools that send MQTT commands to the automation system. The Interaction Agent handles all verbal communication to keep the robot's voice and style consistent. Lastly, the Entertainment Agent suggests activities when it detects that the user has been inactive for a while. We use a fixed routing strategy to manage how these agents work together. Instead of letting the LLM guess which agent to use, we define specific paths in a graph.

4.2. Hybrid Coordination and Prompt Chaining

We moved to a hybrid approach to link high-level reasoning with reliable execution. In earlier versions, we used single, large prompts that contained all the instructions at once. We found that these long prompts were often too much for the smaller LLM to handle, which led to frequent hallucinations and errors. To fix this, we implemented prompt chaining. This method uses a sequence of short, specific prompts instead of one large block of text. Breaking the logic down this way makes the system much more accurate and allows it to handle multiple requests at the same time without getting confused.

```
User: "va a la cocina y enciende la tele"
  ↓
Chain 1: Decompose → 2 tasks identified
  ↓
Chain 2: Prioritize → [1. navigation, 2. TV]
  ↓
Chain 3: Validate → "cocina" is valid ✓
  ↓
Chain 4: Route → Returns ToRobotAgent tool details
  ↓
DIRECT TOOL CALL: Creates AIMessage with ToRobotAgent tool_call
  ↓
Graph executes the REAL tool → Robot actually moves
```

Figure 3: Prompt chaining example

For example, if a user says, "go to the kitchen and turn on the TV", the system goes through several steps as shown in Figure 3. First, it breaks the sentence into two tasks: moving and switching on the appliance. Then, it puts them in order, so the robot moves before trying to use the TV. While behaviour trees could handle similar sequences, they were not as flexible. Because behaviour trees are predefined, they are difficult to modify or extend based on spontaneous user requests. After checking that the kitchen is a valid destination, the system sends the final command using the corresponding agent tools.

4.3. Local Implementation and System Integration

The entire decision-making system is integrated into the ADDIM system using a local edge computing setup. All processing, including the Llama 3.1 8B model running via Ollama, happens on an Intel NUC inside the home. Communication between components relies on a Mosquitto MQTT broker that runs on this same edge computing device. This setup enables the MAS to send instructions to the ADDIM system for robot control and to receive real-time data from the Zigbee sensor network.

5. Results & Discussion

To evaluate the MAS architecture, we tested it in real world domestic settings as part of our ongoing research. Initial trials took place in the private home of an elderly resident in Cartagena, Spain, and in an adapted apartment at the Fundación Poncemar in Lorca, Spain.

5.1. Task Decomposition and Multi-Agent Coordination

During these initial trials, the system showed it could handle natural-language requests with multiple goals quite well. By using the supervisor model, the architecture successfully splits complex sentences into a logical series of tasks. These tasks were then sent to the specific agents responsible for moving the robot, managing the home environment, talking to the user, or suggesting activities.

For example, as seen in Figure 4, when a user asked the robot to "go to the bathroom and open the blinds", the Primary Assistant correctly identified both the movement goal and the environmental action. Standard systems that use static Behaviour Trees would usually need a preprogrammed node for every specific combination of actions like this. In contrast, our MAS architecture executes the sequence on the fly based on what the user actually said.

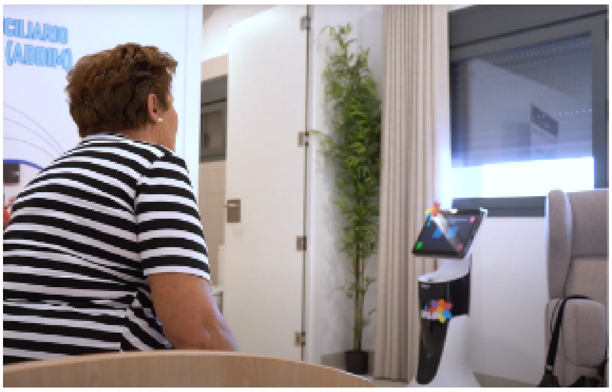

Figure 4: Elderly patient interacting with the robot while the blinds are seen going up in the background

In our preliminary tests, the tasks happened in the right order, and the coordinating agent took back control between each action if required. To quantitatively address system performance, we compared the MAS against the previous Behaviour Tree (BT) baseline in 10 trials of a standard task: instructing the robot to go to the bed and greet the user. As shown in Table 1, the deterministic BT achieved a 100% success rate. In contrast, the MAS had a lower rate due to the LLM occasionally failing to process the request. The table also details system latency. The MAS requires more processing time (35.65 ± 0.55 s) than the BT (1.14 ± 0.18 s). Conversely, tool execution is faster for the MAS (0.14 ± 0.02 s) than the BT (1.16 ± 0.08 s), as the BT includes a programmed 1-second delay for MQTT messaging. Because we used explicit routing and kept the system state separate from the model, we avoided conflicting commands. This meant the robot did not try to adjust the blinds until it had actually reached the correct room, showing that the coordination between different parts of the ADDIM system was reliable.

Table 1: Quantitative Evaluation of Task Execution (N=10 trials)

Task Execution Success Rate		
Component	**BT (Baseline)**	**MAS (Proposed)**
Processing Request	100%	70%
Robot Navigation	100%	70%
User Speaking (TTS)	100%	60%
Statistical Analysis of System Latency (s)		
Latency Metric	**BT (Baseline)**	**MAS (Proposed)**
Request Processing Time	1.14 ± 0.18	35.65 ± 0.55
Tool Execution Time	1.16 ± 0.08	0.14 ± 0.02

5.2. Hardware Performance and Safety

We conducted initial tests of the current system using a mobile robot in real-world settings. The robot's tablet interface acts as a multi-functional portal. Beyond simple navigation commands, users interact with it to perform standardised psychological tests such as the Barthel or STAI scales, play memory games, and follow video-guided physical exercises.

It also provides direct control over the home environment, allowing users to toggle appliances inside the application or open or close motorized blinds. During these first trials, the robot moved between rooms as commanded and the home devices responded correctly to instructions asked by the user to the MAS.

Figure 5 shows one of these sessions, where the user is selecting a psychological test on the robot screen. While these initial results are positive, we still have to do more extensive testing and longitudinal studies to fully validate the MAS.

Figure 5: Elderly user performing psychological tests using the robot's tablet interface

Safety is a primary concern in elderly care, so we implemented confirmation loops for sensitive actions. For example, if the Planning Agent suggests a move that might block a walkway, the Interaction Agent seeks user permission before execution. We also log every decision to allow for detailed analysis of the system logic.

While the supervisor architecture improves predictability, we observed occasional hallucinations during extended conversations. We also found that robot summaries sometimes use technical language that can be confusing for elderly users. Future work will focus on refining the Interaction Agent's prompts to ensure communication is consistently simple and clear.

5.3. Comparison with related works

Table 2 shows how our work compares to other methods in the field. Most traditional systems, such as those reviewed by Iovino et al. Iovino et al. (2022), rely on rule-based engines or Behaviour Trees. These are very reliable for predictable tasks, but they are also very stiff and do not handle the ambiguity of natural speech well. While recent developments like the SDHAR-HOME project by Fidalgo et al. Fidalgo et al. (2025) have automated the generation of these trees using LLMs, they still produce a static execution structure. Our architecture builds on this by allowing the supervisor to coordinate tasks dynamically without needing to generate a full tree for every new request.

On the other hand, many recent research projects use single, large LLM agents. While these are highly flexible, they often struggle with hallucinations and a lack of clear structure in real world environments. Some systems, like the NAO robot interaction described by Casasola et al. Casasola et al. (2025), show great promise for natural conversation but are often limited by their reliance on cloud processing. This dependence, as noted by Anghel et al. Anghel et al. (2020), introduces critical failure points in elderly care due to latency and privacy concerns.

Furthermore, commercial smart speakers like Amazon Alexa or Google Home provide accessible natural-language interfaces that have been shown to improve social engagement and simple environment control Dino et al. (2025). However, as Kim and Choudhury Kim and Choudhury (2021) observed, these devices are limited by their lack of mobility and their "always-listening" nature, which raises significant privacy issues. Unlike these commercial solutions, our system integrates robotic mobility and processes all data locally to ensure both functional autonomy and privacy.

ADDIM-LEAF distinguishes itself from cloud-dependent systems by integrating a local MAS that coordinates social robotics and the smart home environment. Functioning as a proactive health companion, the framework interfaces with the core architecture to manage alerts, verify user status, and adjust environmental controls. By prioritizing task-oriented activities like psychological assessments, the system remains clinically focused, positioning the robot as a facilitator for social engagement rather than a human replacement. While initial results for local multi-intent processing are promising, further work will focus on long-term reliability and hallucination reduction.

The biggest difference between our work and other modern projects is that we run everything locally. Most other LLM-based robots and commercial speakers rely on cloud services, which can be slow and risky for privacy. Our results show that local processing is powerful enough to coordinate complex tasks while keeping personal data safe inside the home. Furthermore, the system continues to work even when the internet is offline or when there is a power outage, as we use an Uninterruptible Power Supply (UPS) for both the router and the NUC.

5.4. Discussion and Qualitative Analysis

The results from the ADDIM system show that organizing tasks into a hierarchy works better than using a single central controller. This is especially true when a user gives vague or "messy" instructions in a real home environment. The supervisor acts as a filter for these requests and keeps track of where the robot actually is, which attempts to stop the LLM from making mistakes based on incorrect information.

Running the Llama 3.1 8B model locally on an Intel NUC involves a tradeoff between privacy and speed. While it keeps the data private, it is not yet fast enough to be considered real time. Some complex tasks can take more than a minute to process from the moment the user speaks to the moment the robot moves. This delay depends on how many sub tasks the system identifies and how much work the agents must do and if hallucinations happen.

5.5. Privacy and Ethics

By processing everything locally, we ensure that sensitive information about a user's life never leaves their house. This is a key part of following GDPR rules and respecting the user's privacy. It also helps elderly users feel more comfortable, as many are worried about devices that are "always-listening" and sending data to the cloud.

To handle the risk of the model making things up, we added a second validation layer. Before the robot moves or changes anything in the house, the system checks the command against a set of safety rules. If the command seems wrong or unsafe, the system switches to a pre-defined safety protocol or asks for help. Combining this with user confirmation makes the robot much safer for a real domestic setting.

5.6. Limitations and Future Work

As this research is still in progress, we identified several areas for improvement during our tests:

- **Context and Scale:** Hallucinations became more frequent during long interactions, likely as the context window reached capacity. The 8B model size also limits reasoning stability compared to larger models. However, experimenting with smaller, faster models remains a priority as their reasoning capabilities continue to improve.

- **Emergency Reliability:** Local processing remains too slow for real-time emergencies. Additionally, potential reasoning errors mean the MAS cannot yet be trusted for important situations. We continue to rely on deterministic, hard coded protocols for immediate emergency responses.

- **Memory Constraints:** Currently, memory is session-based and only exists while a specific request is being handled. The system does not yet retain user preferences or conversational context across different sessions.

- **Task Verification:** The system lacks a robust feedback loop to confirm physical task completion. Agents often assume success once a command is sent, highlighting the need for real-time hardware verification in future versions.

Table 2: Qualitative comparison of the ADDIM architecture with existing AAL and robotic control approaches.

Feature	Deterministic Systems (BT/Rules)	Monolithic LLM Agents	Commercial Smart Speakers	MAS (This Work)
Reasoning Model	Hard coded logic	Generative (End to end)	Cloud based Intent Parsing	Local LLM + Supervisor
Task Flexibility	Low (Predefined)	High (Dynamic)	Medium (Skills/Apps)	High (Orchestrated)
House Integration	Basic/Hardcoded	Often None	With compatible sensors	Full (Unified Core)
Mobility	Variable	High	None (Static)	High (Integrated Robot)
Connectivity	Local	Cloud dependent	Cloud dependent	Fully Local
Privacy Protection	High	Low (Data sent to API)	Low (Always-listening)	High (Edge processing)
Safety Mechanism	Implicit (Fixed)	Probabilistic	Basic confirmation	Explicit (Validation loop)
Interactivity	Limited/Rigid	Natural/Conversational	Natural/Predefined	Natural + Task focused
Fault Tolerance	High (Robust)	Variable (Hallucinations)	Low (No offline support)	Moderate (Local failover)

- **Longitudinal Validation:** While initial trials in Cartagena and Lorca provided positive qualitative data, longer studies are necessary. Future work will gather more extensive feedback from users and clinical staff to assess long-term care impacts.
- **Computing Infrastructure:** Achieving usable local inference speeds currently requires a dedicated Nvidia GPU. Without this acceleration, response times would be impractical. Further optimization is needed to reduce latency as system complexity grows.

Our future work involves broadening the scope of the robot's assistance by integrating several new modules. We plan to add calendar management, message sending, and the ability to manage voice and video calls directly through the robot interface. From a clinical perspective, we aim to include instructed exercise routines, medication reminders, and vision-based detection to confirm that medicine has been taken. We also plan to integrate emotional analysis and conversational pattern analysis, which could help predict early signs of cognitive decline. While these features will run as separate specialized components, they will be coordinated and triggered by the MAS agents to maintain a natural user experience. Additionally, we want to set up a local database for long-term memory so the robot can remember habits and provide more personal experience for the user.

6. Conclusions

This research presents the initial integration of a Multi-Agent System (MAS) into ADDIM-LEAF (Local Edge-based Assistive Framework), demonstrating how an integrated MAS enables assistive systems to move beyond static programming to handle dynamic user needs safely and privately. By transitioning from rigid behaviour trees to a flexible LLM-orchestrated framework, we successfully interpreted complex natural-language requests in real-world settings using prompt chaining for task decomposition. Local edge processing ensured data privacy and system reliability during connectivity failures. As this work is still in progress, upcoming efforts will focus on optimizing response times and extending current capabilities; ADDIM-LEAF establishes a solid, scalable foundation for developing safe, modular decision-making architectures in assistive robotics.

Acknowledgments

The research presented in this paper has been developed within the framework of the ADDIM project (2023–2026), a Public–Private Partnership Project (CPP2022_009649) funded by MICIU/AEI /10.13039/501100011033 and the European Union NextGenerationEU/PRTR. The project is carried out in collaboration with IntecRobots and the Universidad de Murcia as consortium partners.

The authors also acknowledge Fundación Poncemar (Lorca, Spain) and the UPCT–Poncemar Chair for facilitating the use of the adapted apartment for system deployment and evaluation, and for the involvement of the professional therapists who supervised the experimental activities.

References

ADDIM Project, 2025. Addim project. https://www.upct.es/noticias/2025-05-30-testan-en-casa-de-una-nonagenaria-el-sistema-de-apoyo-digital-domiciliario-integral-para-mayores, accessed: 2026-02-26.

Anghel, I., et al., 2020. Smart environments and social robots for age-friendly integrated care services. International Journal of Environmental Research and Public Health 17 (11), 3801.
DOI: 10.3390/ijerph17113801

Blake, H., Jennings, P., Yusuff, M., 2025. Multi-agent ai systems for smart elderly care in assisted living facilities.

Casasola, A. Z., Casanova, E. Z., García-Bermejo, J. G., Domingo, J. D., 2025. Desarrollo de un sistema de interacción mediante agentes llm: Implementación en un robot social nao. Simposios del Comité Español de Automática (CEA) 1 (1).
DOI: 10.64117/simposioscea.v1i1.29

Dino, M. J., et al., 2025. Smart speakers for health and well-being of older adults: A mixed-methods review. Healthcare 13 (21), 2772.
DOI: 10.3390/healthcare13212772

Fidalgo, S. M., Casanova, E. Z., García-Bermejo, J. G., Domingo, J. D., 2025. Generación de behavior trees mediante llm para el control de un robot social. Simposios del Comité Español de Automática (CEA) 1 (1).
DOI: 10.64117/simposioscea.v1i1.10

Iovino, M., Scukins, E., Styrud, J., Ögren, P., Smith, C., 2022. A survey of behavior trees in robotics and ai. Robotics and Autonomous Systems 154, 104096.
DOI: 10.1016/j.robot.2022.104096

Kim, S., Choudhury, A., 2021. Exploring older adults' perception and use of smart speaker-based voice assistants: A longitudinal study. Computers in Human Behavior 124, 106914.
DOI: 10.1016/j.chb.2021.106914

Merino-Fidalgo, S., Sánchez-Girón, C., Zalama, E., Gómez-García-Bermejo, J., Duque-Domingo, J., 2025. Behavior tree generation and adaptation for a social robot control with llms. Robotics and Autonomous Systems 194, 105165.
DOI: 10.1016/j.robot.2025.105165