technologies

MDPI

*Article*

# A Socially Assistive Robot as Orchestrator of an AAL Environment for Seniors †

**Carlos E. Sanchez-Torres, Ernesto A. Lozano, Irvin H. López-Nava *, J. Antonio Garcia-Macias and Jesus Favela ***

Department of Computer Scicence, CICESE, Ensenada 22870, Mexico; hello@sanchezcarlosjr.com (C.E.S.-T.); ernilozpa@gmail.com (E.A.L.); jagm@cicese.mx (J.A.G.-M.)

* Correspondence: hussein@cicese.mx (I.H.L.-N.); favela@cicese.mx (J.F.); Tel.: +52-646-175-0500 (I.H.L.-N & J.F.)

† This paper is an extended version of our paper published in the 15th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2023), Riviera Maya, México, 28–30 November 2023; "An Open Framework for Nonverbal Communication in Human-Robot Interaction", https://doi.org/10.1007/978-3-031-48642-5_3.

**Abstract:** Social robots in Ambient Assisted Living (AAL) environments offer a promising alternative for enhancing senior care by providing companionship and functional support. These robots can serve as intuitive interfaces to complex smart home systems, allowing seniors and caregivers to easily control their environment and access various assistance services through natural interactions. By combining the emotional engagement capabilities of social robots with the comprehensive monitoring and support features of AAL, this integrated approach can potentially improve the quality of life and independence of elderly individuals while alleviating the burden on human caregivers. This paper explores the integration of social robotics with ambient assisted living (AAL) technologies to enhance elderly care. We propose a novel framework where a social robot is the central orchestrator of an AAL environment, coordinating various smart devices and systems to provide comprehensive support for seniors. Our approach leverages the social robot's ability to engage in natural interactions while managing the complex network of environmental and wearable sensors and actuators. In this paper, we focus on the technical aspects of our framework. A computational P2P notebook is used to customize the environment and run reactive services. Machine learning models can be included for real-time recognition of gestures, poses, and moods to support non-verbal communication. We describe scenarios to illustrate the utility and functionality of the framework and how the robot is used to orchestrate the AAL environment to contribute to the well-being and independence of elderly individuals. We also address the technical challenges and future directions for this integrated approach to elderly care.

**Keywords:** ambient assisted living; service orchestration; computational notebook; human–robot interaction; social robots; nonverbal communication

check for updates

Academic Editors: Luc de Witte and Daniele Giansanti

Received: 28 March 2025
Revised: 10 June 2025
Accepted: 13 June 2025
Published: 19 June 2025

**Citation:** Sanchez-Torres, C.E.; Lozano, E.A.; López-Nava, I.H.; Garcia-Macias, J.A.; Favela, J. A Socially Assistive Robot as Orchestrator of an AAL Environment for Seniors. *Technologies* **2025**, *13*, 260. https://doi.org/10.3390/technologies13060260

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Ambient Assisted Living (AAL) environments are smart living spaces designed to enhance the quality of life and independence of elderly or disabled individuals through seamlessly integrated technology. These environments combine sensors, actuators, and intelligent systems to monitor residents' health and activities, provide timely assistance, and ensure safety without being intrusive. By adapting to users' needs and preferences,

AAL environments aim to support aging in place, reduce the burden on caregivers, and promote overall well-being through a combination of proactive and reactive technological interventions [1].

Socially Assistive Robots (SARs) are increasingly considered a component of an AAL. As social robots, they are physical agents designed to communicate with humans while also providing healthcare and caregiving support, such as assisting with activities of daily living, providing companionship, and monitoring health [2].

Given their verbal and nonverbal communication abilities and physical presence, SARs can serve as interactive and intuitive interfaces to the network of sensors and devices present in an AAL environment. By acting as a central hub, these robots can interpret data from various AAL systems, communicate information to users naturally and engagingly, and execute commands to control smart home features, bridging the gap between advanced technology and human-friendly interaction in assisted living spaces.

We present a framework for an adaptable AAL environment that leverages a social robot as a user-friendly interface to smart home services and a computational notebook for customizing the environment by incorporating additional sensors and services. Unlike a basic hub, this framework supports service orchestration, allowing the AAL environment to manage and adapt to complex, dynamic scenarios. It builds on a platform designed to recognize both verbal and nonverbal communication, enabling the social robot to respond appropriately [3], and uses the Eva robot, which has been successfully used to guide interventions with older adults with dementia [4,5].

This work is organized as follows. Section 2 discusses the need for service orchestration in AAL environments. In Section 3 we present the architecture of the framework proposed, including its main components. Section 4 illustrates how the framework composes various services to implement different AAL scenarios. In Section 5 we discuss the novelty of the approach, comparing with service orchestration and the use of social robots in AAL, and present some of the ethical and privacy considerations on the use of the platform. Finally, Section 7 concludes this work and discusses possible future work.

## 2. Orchestration of Ambient Assisted Living Services

AAL environments include many networked sensors, computers, and services aimed at supporting elderly or disabled individuals. Its hardware components usually include motion and pressure sensors, wearable devices, lighting systems, voice assistants, and health monitoring devices. These components support a variety of services such as medication reminders, video conferencing, fall detection, and home automation. As social robots become popular, they are increasingly becoming part of AALs.

As an AAL environment expands with new components and services, it creates opportunities for innovative integrated functionalities. However, this growth also increases the complexity of system management and coordination. For example, integrating a user's smartwatch with sleep detection capabilities could enable the system to postpone medication reminders until the user wakes up, demonstrating how new components can enhance existing services in sophisticated ways.

Voice assistants can function as hubs in smart environments, interfacing with users through voice commands and communicating with various sensors and services. Common examples include turning lights on or notifying users about visitors at the door. These assistants seamlessly integrate multiple smart home functionalities, providing a centralized and intuitive control point for the user.

While software hubs excel at managing connections and data flow, they are limited in handling complex multi-step processes that require sophisticated decision making. Soft-

ware orchestration [6], on the other hand, is designed to manage these intricate workflows, coordinating actions across various systems and adapting to changing conditions.

Recent years have seen an increased interest and research in robotic assistants that can perform tasks focused on social integration, affective bonding, and cognitive training [7]. While verbal communication remains the primary mode of interaction between SARs and humans, non-verbal communication can be particularly effective in AAL environments to trigger services based on the user´s presence, emotional state, or physical behaviors. For instance, facial expression analysis could detect signs of discomfort, prompting an appropriate response; motion sensors detecting an elder standing up in the middle of the night could automatically activate soft lighting for safety; and body posture tracking might trigger fall prevention services. This multimodal approach creates a more intuitive and seamless interaction with the AAL environment.

Social robots can take a variety of shapes and can have different affordances to enact and recognize nonverbal communication cues, showcasing various types of non-verbal human–robot interactions. Some well-known examples include NAO (https://en.wikipedia.org/wiki/Nao_(robot)), accessed on 27 March 2025, which can track faces, make eye contact, imitate human gestures, and respond to touch. It can also exhibit emotions through its LED lights and body movements. The Cozmo (https://www.digitaldreamlabs.com/products/cozmo-robot, accessed on 27 March 2025) robot uses its expressive digital eyes, body language, and sounds to communicate with users, allowing it to exhibit excitement, frustration, and curiosity. Another example is Jibo (https://robotsguide.com/robots/jibo, accessed on 27 March 2025), which uses facial recognition and expressive animations to establish emotional connections with users by exhibiting nonverbal cues like nodding, blinking, and expressing curiosity through head movements. While most of these efforts have focused on sensors within the robot, in an AAL environment, the sensing capabilities can be distributed throughout the living space, allowing for more comprehensive and unobtrusive monitoring of the user's nonverbal cues.

Environmental and wearable sensors, such as cameras, motion detectors, and smartwatches, can work in tandem with the robot's sensors to create a more complete picture of the user's state and behavior. For example, a smartwatch can detect gait patterns and potential fall risks, while ambient microphones can pick up patterns in voice tone that might indicate emotional states. Integrating robot and environmental sensing enables a more context-aware and responsive system capable of interpreting subtle nonverbal cues that a robot alone might miss.

Moreover, this expanded sensing network allows the social robot to act as an intelligent interface to the broader AAL system, translating environmental data into appropriate social responses and actions. This synergy between the robot and the environment enhances the capability for nonverbal communication and creates a more seamless and natural assisted living experience for the user.

### *2.1. Nonverbal Interactions in HRI*

Human–robot interaction can incorporate various forms of nonverbal communication in an AAL environment. These interactions utilize a broad spectrum of communication modalities and cues that go beyond spoken language [8]. The following are some categories of nonverbal interactions in AAL scenarios:

1. Visual cues: Analyze a person's facial expressions, such as smiles, frowns, raised eyebrows, or narrowed eyes, to infer emotions like happiness, sadness, surprise, or anger; this also includes eye gaze tracking and whether a person is making or avoiding direct eye contact. All these might be signs of interest or discomfort.

2. Paralanguage: Tones of voice, such as volume and speed, can indicate mood, exertion, or health status. Detecting non-speech vocalizations such as sighs or groans might indicate stress or pain.
3. Gestures: Hand movements or arm motions to communicate with humans. Gestures can include pointing to objects, waving, making other meaningful movements to indicate actions and directions, or engaging in collaborative tasks.
4. Body Language and Posture: By observing a person's posture and overall body movements, a robot can interpret cues like open or closed body position, leaning in or away, hand movements, gait analysis or crossed arms, providing insights into the person's level of comfort, engagement, or risk.
5. Proxemics: Understanding the distance between the robot and the person using sensors, such as the camera, allowing the robot to assess the person's comfort level with proximity and adjust its behavior accordingly.
6. Time-based cues or chronemics: Patterns of user activities, including time to respond to robot interactions, day–night activities, and chrononutrition.

These types of nonverbal interactions are crucial in making human–robot interactions more natural, intuitive, and socially engaging.

### *2.2. Design Scenarios*

We describe three scenarios of human–robot interaction in the AAL environment that were developed to identify the design requirements for the proposed framework. These scenarios highlight three main roles social robots can play in ALL: companionship and emotional support, communication facilitation, and health monitoring. They are derived from design sessions with caregivers [9] and proven applications of social robots in elder care [10].

#### 2.2.1. Scenario 1: Guiding an Exercise Routine

Elsa is an older adult who performs her exercise routine each morning. The robot activates the physiological sensor in Elsa's smartwatch to continuously measure her heart rate when she starts her exercise routine. One morning, while exercising, Elsa's heart rate increases beyond a predefined threshold, which is considered dangerous or inappropriate for her health. Thus, the social robot will issue a verbal warning to advise her to stop or decrease the intensity of the exercise to avoid health risks.

#### 2.2.2. Scenario 2: Gaze and Proxemics to Initiate an Interaction

Eva is a social robot used to assist Roberto, a patient who lives alone at home and exhibits signs of dementia. He often feels disoriented and lonely. When feeling like talking to someone, he approaches Eva to get its attention and initiate an interaction. As he approaches her, the robot can analyze and calculate how far Roberto is from it using an integrated camera. If the distance between them is greater than three meters, Eva will begin to move her head to track him. As Roberto approaches while looking at the robot, the robot will change its behavior to be more interactive and affective. When Eva notices that Roberto keeps looking at it for more than two seconds, the robot will initiate an interaction: "Hi Roberto, how have you been today?", and the conversation will continue, incorporating elements of cognitive stimulation therapy until Roberto decides to terminate the interaction.

#### 2.2.3. Scenario 3: Sleep Disturbances

Ana usually sleeps soundly, but tonight, after three hours of rest, she becomes restless. Eva cannot determine a probable cause by analyzing diurnal sleep patterns, environmental temperature, noise, or Ana's food and water intake that day. At 2 a.m., Eva detects that

Ana is awake based on her smartwatch's heart rate and accelerometer data. Eva attempts to comfort and encourage Ana to go back to sleep by performing a mindfulness activity. However, Ana stands up and tells Eva that she must use the bathroom. Responding to Ana's movements, detected by motion sensors, Eva activates soft lighting at the base of Ana's bed to guide her safely to the bathroom. Ana goes to the toilet twice that night, which is unusual for her. Unable to identify a clear cause for this behavior, Eva informs a caregiver to check for a possible urinary tract infection. Meanwhile, Eva reminds Ana the next day not to drink too much water before bedtime and adjusts the room temperature to an optimal level for sleep.

## 3. Architecture of the AAL Orchestration Framework

We present the proposed AAL orchestration architecture, focusing on its features rather than implementation details, from a domain perspective. In this context, the architecture's competence refers to the domain it operates in (e.g., nonverbal scenarios, behaviors, and tasks). In contrast, its performance depends on implementation factors such as technology choices, algorithms, and design practices.

Our architectural framework is a flow-based programming engine that orchestrates microservices through streams. Inspired by [11], our proposal emphasizes distributed and data-driven strategies. These are characteristics that form the basis for our proposed framework and that we could not find in already existing platforms. For example ROS [12], the most widely used platform, consists of a set of software libraries and tools to build robot applications; as flow-based features are not present, third-party proposals (such as [13,14]) have recently appeared.

An AAL environment relies on distributed algorithms running across multiple interconnected nodes without strict centralized control, and online algorithms (including stream and dynamic algorithms), which process input incrementally rather than all at once, resembling nonmonotonic logic. In addition, reactive programming provides operational semantics that respond to external events and seamlessly manage time-dependent data flows, such as asynchronous environmental signals.

The proposed engine architecture is event-driven and utilizes a combination of publish/subscribe models through various communication protocols with different fault-tolerance mechanisms, along with a computational notebook that enables users to create reactive microservices. In this context, an event is defined as an immutable object, denoted as the evaluation of the signal, $s(t_0)$, or the change in the state of the system, originating from a subsystem. In discrete time increments, this object yields a stream, the signal $s(t)$. A producer generates an event that multiple consumers can process.

The messaging system operates in one of three ways: (i) producers send direct messages without intermediaries, (ii) messages are routed through a message broker, or (iii) events are logged for later processing. In this work, we refer to both producers and consumers as microservices—independently deployable and loosely coupled, unlike monolithic software.

Once the consumer receives a stream, several processing approaches can be applied:

- Data synchronizers store or present data in a storage system.
- Tasks trigger events that influence the environment.
- Pipeline stages aggregate events from input streams, producing one or more output streams that pass through various transformations before reaching a task or data synchronizer via pipe-and-filter operations.

Thus, a service forms a graph of dependencies among microservices, connected by different protocols, working together to achieve a specific competence.

What can be done with the stream once it is in the hands of the consumer? How can the consumer process it and when can we achieve a specific competence? Those are classified as data synchronizers, tasks, and pipeline stages. Data synchronizers are tasked with storing or presenting data on a storage system. Tasks push events that affect the environment. Pipeline stages receive events from multiple input streams to produce one or more output streams that go through different stages until they reach a task or data synchronizers applying pipe-and-filter operations. Therefore, a service is a graph of dependencies within microservices joined by different protocols designed to achieve a specific competence.

In contrast to other alternatives, the process decision mechanism introduces a perspective of fault tolerance in microservices. At the worker level, we implement a "let it crash" policy with a supervisor.

Figure 1 illustrates the data flow within an example of a service, where the nodes symbolize microservices and the edges represent communication protocols. A real-time conversation, which is a specific competence, with a social robot involves understanding communication to either wake up or put the system to sleep. Instead of using a hotword to wake up the assistant, the rest of the system remains largely inoperative until sufficient evidence is obtained to make a decision. In addition, a conversation involves the user managing sessions, where the social robot can recall the voice, face, and other characteristics of the user to interact in the best way customized for that user. When EvaNotebook, a component designed to build microservices on the fly, wakes up the system based on available information, it customizes the Large Language Model (LLM) to respond based on a specific scenario by injecting a system prompt and restoring previous sessions. The Rhasspy is then used to vocalize the robot's response and listen to the user's intention. Whisper transforms speech into text and sends it back to EvaNotebook. Therefore, we form a feedback loop where different consumers emit events that we must inject into the LLM and react accordingly. In the background, the system records data for later analysis.

The remainder of this section focuses on the physical and logical components needed to build AAL services, including the social robot, wearable and environmental sensors, an MQTT Broker, and EvaNotebook.

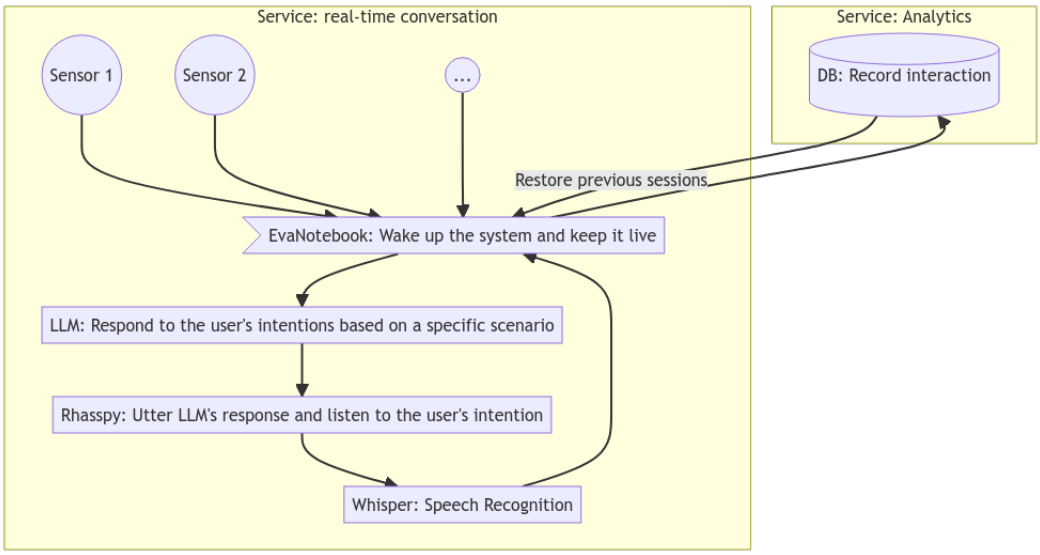



**Figure 1.** A typical topological data flow in our event-driven architecture.

### *3.1. The Eva Social Robot*

EVA is an open-hardware robotic platform (https://eva-social-robot.github.io/, accessed on 27 March 2025) characterized by its modular design and low cost. It incorporates

several sensors and supports verbal and nonverbal communication. The platform was initially developed as a social assistant robot to interact with and assist people with dementia [4]. EVA has been used in other scenarios, such as helping children with emotional regulation [15].

### 3.2. EvaNotebook

EvaNotebook (https://notebook.sanchezcarlosjr.com, accessed on 27 March 2025) is a computational notebook designed to operate within a browser environment without needing a client–server architecture. Its design lends itself to developing prototypes, experimental processes, and system scenarios, mainly emphasizing multilingual programming for event-driven architectures. It achieves this with the assistance of a decentralized database and incorporates various application protocols, including WebRTC, WebSockets, and MQTT, providing a higher level of abstraction. Our notebook extends the Jupyter Notebook v4.5 JSON schema, introducing a block-based application framework incorporating dynamic dependencies and configurations through an advanced plugin architecture delineated in the metadata. Each code block within our system operates based on an actor model with a supervision tree model similar to Elang. In the actor model, a code block can send a limited number of messages, spawning a finite number of new actors and defining its behavior for processing incoming text messages. In this approach, each worker is executed on different nodes, adopting a DAG (Directed Acyclic Graph) execution model instead of a linear model. This method is advantageous because it is often more practical to move the code to where the data and other critical dependencies reside rather than relocating the data itself.

### 3.3. MQTT Broker

We have adopted a pub-sub architecture with loosely coupled components, allowing us to scale and adapt services more quickly than a centralized architecture. This approach enables us to externalize the processing and view the social robot and the rest of the environment as both subscribers and emitters of events. As a communication protocol to distribute these events, we use MQTT, the current standard protocol for IoT, a lightweight messaging protocol designed for efficient communication in constrained environments. The platform deploys an instance of Eclipse Mosquitto (https://mosquitto.org/, accessed on 27 March 2025), an open-source message broker that implements the MQTT protocol on a local network to guarantee low latency and ensure responsiveness.

### 3.4. Sensors

Several sensors integrated into social robots like Eva, in the user and/or the environment, are used to gather data. These sensors provide valuable insights into human emotions, physical cues, and environmental context by capturing and analyzing data. Eva can leverage this information to deliver personalized and adaptive interactions, fostering various benefits for individuals. Some of the advantages that this offers to multiple interactions are:

- Emotional understanding. Eva can perceive a person's emotional state and respond accordingly by analyzing physiological signals and facial expressions.
- Contextual adaptation. Environmental sensors, including microphones and cameras, allow EVA to sense and respond to the environment around it.
- Supportive and assistive capabilities. The framework processes data to offer support to individuals. For instance, if Eva detects an increase in heart rate or signs of stress through wearable sensors, it can trigger relaxation techniques, breathing exercises, or soothing music to alleviate anxiety.

- Data sharing and integration. Eva's ability to deliver information captured by these sensors via MQTT to other devices or algorithms opens up further analysis and integration opportunities.

### *3.5. Voice Assistant*

For verbal interaction with the user, the framework uses Rhasspy (https://rhasspy.readthedocs.io/en/latest/, accessed on 27 March 2025), an open-source toolkit for creating personalized voice assistants with a strong focus on user privacy. It enables developers to build custom voice interfaces for controlling devices and services through MQTT. A standout feature of Rhasspy is its commitment to data protection. By default, the system is designed to function completely offline, ensuring that all voice processing occurs within the user's local network. This approach eliminates the need to transmit voice data to external servers, thereby safeguarding user privacy [16].

### *3.6. Data Analysis Models*

The framework can use pre-trained gestures and emotions in posture recognition modules to infer relevant activities, behaviors, and nonverbal cues. Several EvaNotebooks have been developed to recognize gaze direction, proximity, object recognition, posture, etc. These can be adapted or customized to the problem and run synchronously with other tasks.

## 4. Implementing Assistive Services

In this section, we illustrate how the services described in the scenarios presented in Section 2.2 are implemented using the orchestration paradigm supported by the framework.

### *4.1. Scenario 1: Guiding an Exercise Routine*

Figure 2 shows a sequence diagram to explain the interaction of Scenario 1, described in Section 2.2.1. The social robot Eva initiates the interaction by indicating to the elder the exercise routine to be performed. The robot initiates a service in EvaNotebook to be notified when the user's heart rate deviates from established parameters. When this happens, the microservice in EvaNotebook will warn the user so that he adjusts the routine, advises the user to take a break, and/or triggers a notification to a third party of the situation.

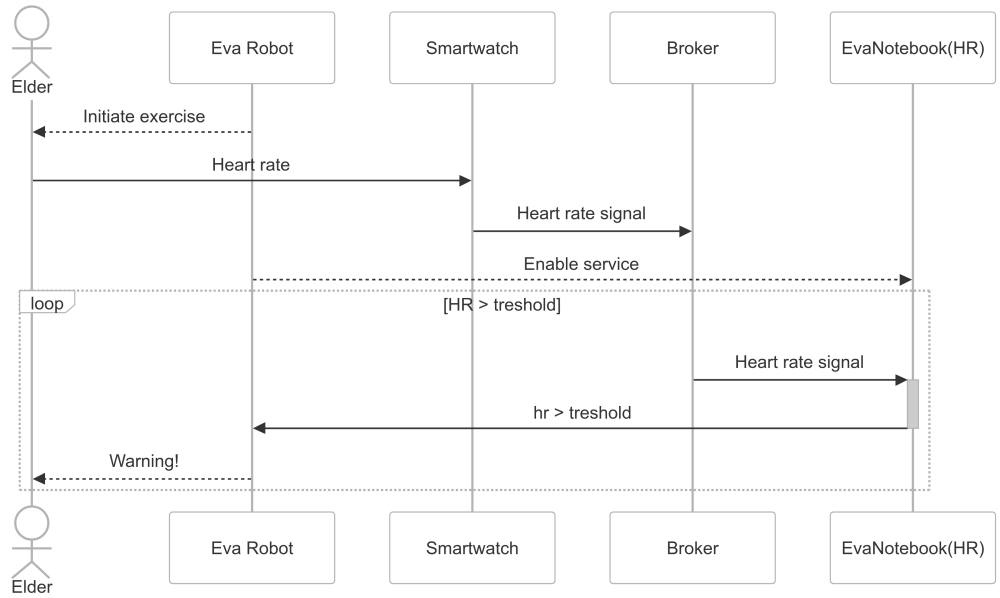



**Figure 2.** Eva guides the exercise routine of the user, monitoring his heart rate.

This scenario requires access to data from sensors that are not integrated into the robot but worn by the subject. The components of the proposed framework operate together to accomplish the task.

1. Adding a sensor. The device with the physiological sensor is added to the environment by registering with the broker, and a JSON specification is used to indicate the frequency of the data sent to the broker. The user wears a sensor that monitors their heart rate during the exercise routine, such as a smartwatch with an optical ECG or a chest strap monitor with ECG electrodes. Both types of monitoring devices can be connected via Bluetooth. Sensors requiring synchronous communication, such as video, can be connected directly to the consumer.
2. Data collection. A service coded in a notebook for processing ECG data registers to the broker when executed. By using the Pulsoid API, the data can be transferred to the EvaNotebook for processing.
3. Data processing. A script in EvaNotebook is used to analyze the incoming data, which can be plotted in real-time for visualization purposes. If the heart rate parameters are out of range, a message can be enabled with Rhasspy. The EvaNotebook can also connect to the SendGrid API, which provides the necessary tools for sending notifications via email and text. This functionality is designed to send alert notifications to a caregiver or an individual associated with the user.

### *4.2. Scenario 2: Gaze and Proxemics to Initiate an Interaction*

In this scenario, the social robot Eva initiates an interaction when it detects the presence of an individual at a certain distance. It uses the camera and a facial recognition algorithm to infer gaze direction and proximity.

This is implemented in an EvaNotebook that extends the "WakeFace" microservice proposed in [17] as an alternative to the wakeword method to initiate an interaction. The extension consists of using proximity in addition to gaze direction to adapt the interaction according to the following criteria:

1. Distance greater than three meters: When Eva detects a person walking at a distance of more than three meters, the robot activates its visual tracking function. Eva turns its neck and directs its gaze toward the person as long as the person keeps moving. The robot adjusts its neck's speed and range of motion to follow the person's movement smoothly and naturally. However, Eva does not initiate verbal interaction or ask questions at this distance.
2. Distance less than three meters: When the person comes within approximately three meters of Eva, the robot emits a soft auditory signal to capture the person's attention and then initiates a verbal interaction. For example, it might say, "Hello! How are you today?" or "Welcome! Can I help you with something?"
3. The service of a real-time conversation: Once Eva has initiated the conversation, it waits for the user's response. If the person responds positively or shows interest, Eva continues the dialogue smoothly and naturally, enacting conversation-based therapy using an LLM with a prompt tailored to the individual and relevant context [18].

The code below is an extract from the WakeFace microservice written in EvaNotebook, specifically in a WebWorker of JavaScript. It illustrates how the microservice subscribes to relevant MQTT events. The obtained camera data is then sent to the ML modules responsible for assessing gaze direction and proximity (MediaPipe). The code uses a pipe-and-filter architecture to handle asynchronous events, allowing for streaming processing via operators. The code listens to the camera stream, ignoring frames where no face is detected. When a face is detected, it tests if the user is facing the camera and within 3 m; if so, it runs the wakeup script to greet the user and initiate a conversation. The use of a

pipe-and-filter approach facilitates adding complexity to the service, for instance, by using a model to infer the mood of the user and modify the greeting script accordingly. If the facial emotion recognition model is available, this would imply adding a line of code to the script. Additionally, this service is in itself a notebook that can be used as a component by more sophisticated services.

**Listing 1.** WakeFace microservice that runs on Evanotebook.

```
begin
   // import mediapipe
   zip(
    listen('Mqtt', 'camera stream').pipe(
      map(frame => mediapipe.faceDetector.detect(frame).detections[0]),
      // Ignore frames when no face detected.
      filter(landmarks => landmarks !== undefined),
      // Check if the user is facing the robot
      map(landmarks => testIfUserLooksAtCamera(landmarks))
    ),
     listen('Mqtt', 'camera stream').pipe(
      map(frame => mediapipe.faceLandmarks.detect(frame).detections[0]),
      filter(landmarks => landmarks !== undefined),
      map(landmarks => testIfUserIsADistanceLessThan3Meters(landmarks))
     )
   ).pipe(
       filter(
      // Run wakeup script if user close and facing the robot
        ([userLooksAtCamera, userIsAdistanceLessThan3Meters]) =>
             userLooksAtCamera && userIsAdistanceLessThan3Meters),
       tap(_ => mqtt.publish('wakeup'))
   )
end
```

Mediapipe is used to detect face landmarks. Afterward, the code estimates whether the person is looking at Eva. Different approaches to achieve this include solving PnP, Procrustes Analysis, SVD, or defining an ROI. On the other hand, we calculate the user's distance to the camera using, for instance, the pinhole camera model. Then, we combine both streams to determine whether to wake up the system.

Once the conversation starts, EvaNotebook prompts ChatGPT using its API. The prompt describes the scenario, such as reminiscence therapy or cognitive stimulation. Then, it indicates general recommendations for talking with people with dementia, such as using short sentences and avoiding criticizing and correcting. Then, the user's profile and preferences are provided, and finally, a sample conversation.

### *4.3. Scenario 3: Sleep Disturbance*

The scenario on sleep disturbance mentioned above can help highlight several of the technical advantages offered by EvaNotebook and the orchestration framework:

1. Reactive Programming. EvaNote offers transparent layers of inter-communication tools in a distributed, heterogeneous computing environment, inspired by Marimo, ViNE, and the principles of Functional Reactive Programming [19,20]. EvaNote presents a reactive programming environment that guarantees transparency for developers in distributed settings. When a code block is executed, EvaNote promptly reacts by automatically triggering the execution of subsequent cells that rely on the variables defined in the initial block. This functionality streamlines the workflow, eliminating the tedious and error-prone process of manually re-executing dependent cells. When a cell is removed, EvaNote proactively purges its variables from the program's memory using a Garbage Collector, thereby avoiding any hidden states that could lead to

unpredictable results. In this scenario, when a condition is met (person wakes up), the AAL environment reacts by triggering an appropriate response for a given context (time = 2 a.m.) to encourage Ana to get back to sleep or contact a caregiver.

2. Service Coordination. A notebook includes code blocks that define one or more services and are deployed independently of other services leveraging a mobile agent strategy, allowing them to self-replicate and use metadata to determine their execution context autonomously—be it within a computing cluster, as a standalone microservice, or on a blockchain framework. Each code block in EvaNotebook operates based on either an actor model or a shared memory paradigm. The execution environment is server-agnostic. The process decision mechanism introduces a different perspective for the notebook, adopting a DAG (Directed Acyclic Graph) execution model instead of a linear model. In this approach, each block is executed on different nodes. This method is advantageous because it is often more practical to move the code to where the data and other critical dependencies reside rather than relocating the data itself. In the sleep scenario, the service that monitors night behaviors depends on the sleep data and other relevant data that could cause abnormal behavior (gender, frequency of visits to the bathroom, water intake, room temperature).
3. Data Storage. As a Distributed File System, EvaNotebook uses "EvaFS," a specialized metadata layer designed to function as file system middleware by positioning itself atop existing storage infrastructures. EvaFS enhances the functionality of underlying storage systems by managing metadata, thereby facilitating more sophisticated data operations and interactions without directly handling data storage. The metadata produced by a sensor or a service, such as HR or accelerometer raw data, is described in JSON.
4. End-user Programming. End-user development enables users, particularly those without formal programming skills, to create, modify, or enhance software without necessitating code recompilation [21]. This approach integrates the low-code paradigm with techniques like programming by example and visual programming. EvaNotebook supports end-user programming via a flexible architecture powered by the virtualization of WebComponents, which allows the dynamic modification of the playbook executor across various nodes. The user can modify a service if, for instance, a new device is added to the environment, such as indicating that the new under-bed lights should be turned on when the older adult stands up from bed at night.
5. Collaboration. EvaNotebook supports seamless real-time collaboration through a transparent mechanism replicating the working environment. Its security infrastructure is underpinned by an Identity and Access Management (IAM) system, coupled with a secure storage solution known as Vault, which ensures secure user authentication and safeguards sensitive data. In the scenario above, a caregiver can access some or all of the notebooks in the AAL environment to monitor variables or edit a service directly. For instance, it could edit a sleep service to set the thermostat to 18 degrees Celsius when the user goes to bed.

## 5. Ethics and Privacy Considerations

Introducing social robots in elderly care scenarios requires careful consideration regarding privacy and ethical issues. Integrating AI in assistive robots introduces concerns regarding collecting, storing, and utilizing user data; SARs often use cameras, microphones, and sensors to interact with users, which can lead to constant monitoring and a loss of personal privacy. As described in Section 3.5, we handle locally the transmission and processing of data related to voice interactions with the robot without external servers or communication links; this includes using MQTT over SSL to enable secure communications.

Although we have taken measures to protect data transmission, it is still necessary to implement secure data storage measures (including encryption and access controls) to protect stored data from unauthorized access or breaches. Other measures we intend to incorporate into future system versions include techniques to anonymize or de-identify data whenever possible, thus safeguarding user identities.

Evaluation of the system with actual users will require obtaining explicit consent from users for data collection and usage, providing them with the ability to access, modify, or delete their data; this should include establishing clear data retention policies, specifying the duration of data storage and procedures for secure deletion. If actual (elderly) users are not capable of giving informed consent (for example, in cases of dementia or other cognitive impairments), it may be necessary to involve family members or legal guardians. Informed consent is a process that we have duly followed in all interventions (e.g., [4,5,9]) where we have used the same SAR system.

It is also essential to remember that the primary ethical consideration is to guarantee the safety and well-being of the elderly interacting with the robot. Therefore, measures should be taken to protect them from physical, emotional, and psychological harm. In case of error or malfunction (e.g., if a robot fails to alert for a fall), it is critical to establish clear accountability guidelines in order to define the responsibilities of system developers, caregivers, and care institutions.

## 6. Discussion

A thorough examination of the literature does not yield many instances where social robots act as services orchestrators in AAL settings. We can find frameworks that apply service orchestration to robot application development [22].

Within the healthcare sector, hospOS is a platform for orchestrating service robots in hospitals, aimed at alleviating the shortage of healthcare workers [23]. Similarly to our approach, the design of hospOS is informed by scenarios or case studies. However, their approach is on the autonomous navigation of the service robot triggered by a workflow request made by the staff on a web application rather than the end user communicating directly with the robot, as required in an AAL environment.

The need to orchestrate services has been recognized in the context of ambient assisted living. An experimental ambient orchestration system was built on top of the universAAL middleware [24] to address the challenge of interoperability of various commercial devices [24]. The underlying technology is based on web services. This approach is similar to the one we propose in the sense that it supports the interconnection of diverse devices that are registered as nodes in the universAAL middleware. However, the emphasis is on inter-device interaction, rather than the role of the user, which in our proposal relies on human–robot interaction.

Similarly, in [25], the authors propose an ontology that supports the creation and orchestration of services in an AAL environment. The approach also focuses on the problem of the technical integration of diverse devices, rather than their seamless interaction with the user.

The MPOWER project focuses on modeling user needs to support the orchestration of AAL services [26]. It describes two pilot scenarios that prioritize service automation and care provider interaction with the system, rather than requiring the elderly to initiate orchestration as we propose.

The use of SARs has shown potential for emotional connection, as in the case of CuDDler [27], where a participant with visual agnosia formed a strong emotional bond with the robot through tactile interaction, linking it to memories of her dog, thus highlighting the variability of responses and the potential therapeutic benefit for select individuals.

On the other hand, the role of social robots is increasingly being explored as part of AAL systems [28]. This includes predicting user activities to provide proactive assistance [29], monitoring air quality and triggering alerts [30], monitoring and providing social and cognitive stimulation to older adults [31], and assisting in performing household tasks [32]. While users interact directly with these robots, the services provided are usually independent of each other and predefined in advance, and do not require the adaptive orchestration of devices and services.

The proposed approach facilitates the development of social robots that have demonstrated utility in older adult care, but might require tailoring to the individual or context of use. These applications include guiding exercise routines [10], leading therapy sessions for individuals with dementia [4], and providing mealtime assistance to those with dementia [5].

The proposed orchestration architecture seeks to bridge the gap between programming effort and service complexity in social robots for elderly care. While graphical programming languages offer an intuitive, end-user-friendly approach [33], they often lack the flexibility and expressiveness needed to handle intricate, real-world scenarios. Conversely, XML-based or other text-heavy programming methods provide the necessary control for complex tasks [34], but are too cumbersome for non-technical users. This architecture strikes a balance by combining high-level abstractions, in the form of notebooks that implement atomic services, that simplify service creation with underlying modular components capable of managing sophisticated interactions. By doing so, it empowers caregivers and other end-users to design and customize robot behaviors without requiring advanced programming skills, while still accommodating the nuanced demands of elderly care environments. The result is a scalable, adaptable framework that democratizes robot programming without sacrificing functionality.

Although the scenarios used to demonstrate the orchestration platform have not been validated with actual users, we have previously conducted interventions using the Eva robot to guide therapy sessions for individuals with dementia [4] and supporting older adults with eating disorders [5]. Additionally, other applications, such as employing a social robot as an exercise coach for older adults, have been successfully implemented [10]. Therefore, our focus is on enabling the deployment of such services using a socially assistive robot (SAR), rather than on assessing user adoption or the clinical efficacy of the interventions themselves.

Our proposed approach centers on human–robot interaction, creating an intelligent interface that enables end-users to seamlessly orchestrate devices and services toward both explicit and implicit goals—a novel approach not previously documented in existing research literature.

## 7. Conclusions and Future Work

This paper presented a novel framework for integrating social robotics with Ambient Assisted Living technologies to improve elderly care. The proposed approach leverages the unique capabilities of social robots as central orchestrators in AAL environments, combining their emotional engagement abilities with comprehensive monitoring and support features. By using a social robot as an intuitive interface to complex smart home systems, developers can create a more accessible and user-friendly environment for seniors and their caregivers.

Implementing a computational P2P notebook for environment customization, reactive services, and machine learning models for non-verbal communication recognition enhances the system's adaptability and responsiveness to individual needs. The human–robot interaction scenarios presented illustrate this framework's practical applications and benefits in real-world settings.

Two key components of the framework are a broker server that distributes messages from sensors, actuators, and analysis modules, and the P2P computational network used to quickly develop and test prototypes that can become components of more complex scenarios. We illustrate the use of the framework with multimodal scenarios that use various sensors and inference models.

Future work includes exploring different perspectives on failure tolerance, as relying solely on a single worker's perspective is insufficient. Implementing Collaborative Filtering or Graph Machine Learning with Consensus algorithms is an effective strategy to enhance fault tolerance and decision making.

The current paradigm of end-user programming involves writing or editing scripts in a notebook, which might be too complex for most users. We plan to add an automatic programming feature supported by an LLM using prompt-tuning and the Retrieval Augmented Generation paradigm. The documentation of new services, including examples of their use, will be added as an external context. Thus, the user could create a new notebook using instructions in natural language such as, "Ask me if I took my medication when I go to bed if I have not already done that."

As the global population ages, innovative solutions like the one proposed become increasingly crucial. By combining the strengths of social robotics and AAL technologies, we can create more effective, efficient, and compassionate care environments for our elderly population. This integrated approach not only has the potential to revolutionize elderly care but also to set new standards for human–robot interaction in assistive contexts.

**Author Contributions:** Conceptualization, I.H.L.-N. and J.F.; methodology, C.E.S.-T., I.H.L.-N. and J.F.; software, C.E.S.-T. and E.A.L.; validation, E.A.L., C.E.S.-T. and J.A.G.-M.; writing—original draft preparation, J.F., C.E.S.-T.; writing—review and editing, I.H.L.-N. and J.A.G.-M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Mexican Ministry of Science, Humanities, Technology and Innovation (SECIHTI) thru scholarships provided to the first two authors.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Blackman, S.; Matlo, C.; Bobrovitskiy, C.; Waldoch, A.; Fang, M.L.; Jackson, P.; Mihailidis, A.; Nygård, L.; Astell, A.; Sixsmith, A. Ambient assisted living technologies for aging well: A scoping review. *J. Intell. Syst.* **2016**, *25*, 55–69. [CrossRef]
2. Abdi, J.; Al-Hindawi, A.; Ng, T.; Vizcaychipi, M.P. Scoping review on the use of socially assistive robot technology in elderly care. *BMJ Open* **2018**, *8*, e018815. [CrossRef]
3. Lozano, E.A.; Sánchez-Torres, C.E.; López-Nava, I.H.; Favela, J. An Open Framework for Nonverbal Communication in Human-Robot Interaction. In Proceedings of the 15th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2023), Lima, Peru, 22–24 November 2023; Bravo, J., Urzáiz, G., Eds.; Springer: Cham, Switzerland, 2023; pp. 21–32.
4. Cruz-Sandoval, D.; Morales-Tellez, A.; Sandoval, E.B.; Favela, J. A Social Robot as Therapy Facilitator in Interventions to Deal with Dementia-Related Behavioral Symptoms. In Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, New York, NY, USA, 23–26 March 2020; HRI '20; pp. 161–169. [CrossRef]
5. Astorga, M.; Cruz-Sandoval, D.; Favela, J. A Social Robot to Assist in Addressing Disruptive Eating Behaviors by People with Dementia. *Robotics* **2023**, *12*, 29. [CrossRef]
6. Thomas, E. *Service-Oriented Architecture: Concepts, Technology and Design*; Prentice Hall: Upper Saddle River, NJ, USA, 2005.
7. Robinson, H.; MacDonald, B.; Broadbent, E. The role of healthcare robots for older people at home: A review. *Int. J. Soc. Robot.* **2014**, *6*, 575–591. [CrossRef]
8. Urakami, J.; Seaborn, K. Nonverbal cues in human–robot interaction: A communication studies perspective. *ACM Trans. Hum.-Robot Interact.* **2023**, *12*, 1–21. [CrossRef]

9. Cruz-Sandoval, D.; Tentori, M.; Favela, J. A Framework to Design Engaging Interactions in Socially Assistive Robots to Mitigate Dementia-Related Symptoms. *J. Hum.-Robot Interact.* **2024**, *14*, 3700889. [CrossRef]
10. Cauli, N.; Reforgiato Recupero, D. Dr.VCoach: Employment of advanced deep learning and human–robot interaction for virtual coaching. *Intel. Serv. Robot.* **2025**, *35*, 123–135. [CrossRef]
11. Pineda, L.A.; Rodríguez, A.; Fuentes, G.; Rascon, C.; Meza, I.V. Concept and functional structure of a service robot. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 6. [CrossRef]
12. Quigley, M.; Gerkey, B.; Smart, W.D. *Programming Robots with ROS: A Practical Introduction to the Robot Operating System*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2015.
13. Wang, S.H.; Tu, C.H.; Huang, C.C.; Juang, J.C. Execution Flow Aware Profiling for ROS-based Autonomous Vehicle Software. In Proceedings of the Workshop Proceedings of the 51st International Conference on Parallel Processing, Chicago, IL, USA, 15–18 August 2022; pp. 1–7.
14. Lumpp, F.; Panato, M.; Bombieri, N.; Fummi, F. A design flow based on docker and kubernetes for ros-based robotic software applications. *ACM Trans. Embed. Comput. Syst.* **2024**, *23*, 1–24. [CrossRef]
15. Rocha, M.; Valentim, P.; Barreto, F.; Mitjans, A.; Cruz-Sandoval, D.; Favela, J.; Muchaluat-Saade, D. Towards Enhancing the Multimodal Interaction of a Social Robot to Assist Children with Autism in Emotion Regulation. In *Pervasive Computing Technologies for Healthcare*; Lewy, H., Barkan, R., Eds.; Springer: Cham, Sitzerland, 2022; pp. 398–415.
16. Filipe, L.; Peres, R.S.; Tavares, R.M. Voice-activated smart home controller using machine learning. *IEEE Access* **2021**, *9*, 66852–66863. [CrossRef]
17. Villa, L.; Hervás, R.; Cruz-Sandoval, D.; Favela, J. Design and evaluation of proactive behavior in conversational assistants: Approach with the eva companion robot. In *International Conference on Ubiquitous Computing and Ambient Intelligence*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 234–245.
18. Favela, J.; Cruz-Sandoval, D.; Parra, M.O. Conversational Agents for Dementia using Large Language Models. In Proceedings of the 2023 Mexican International Conference on Computer Science (ENC), Guanajuato, Mexico, 11–13 September 2023; pp. 1–7.
19. Perez, I.; Dedden, F. The Essence of Reactivity. In Proceedings of the 16th ACM SIGPLAN International Haskell Symposium, Berlin, Germany, 21–22 August 2023; pp. 18–31.
20. Kleppmann, M.; Wiggins, A.; Van Hardenberg, P.; McGranaghan, M. Local-first software: You own your data, in spite of the cloud. In Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, Portland, OR, USA, 21–23 October 2019; pp. 154–178.
21. Barricelli, B.R.; Cassano, F.; Fogli, D.; Piccinno, A. End-user development, end-user programming and end-user software engineering: A systematic mapping study. *J. Syst. Softw.* **2019**, *149*, 101–137. [CrossRef]
22. Qi, L.; Zhang, X.; Chen, H.; Bian, N.; Ma, T.; Yin, J. A Novel Distributed Orchestration Engine for Time-Sensitive Robotic Service Orchestration Based on Cloud-Edge Collaboration. *IEEE Trans. Ind. Inform.* **2025**, 45, 215–229. [CrossRef]
23. Schmidt, S.; Sommer, D.; Greiler, T.; Wahl, F. hospOS: A Platform for Service Robot Orchestration in Hospitals. In Proceedings of the ICT4AWE, Vienna, Austria, 9–11 April 2024; pp. 221–228.
24. Brestovac, G.; Marin, D.; Oroz, T.; Vidović, A.; Bozóki, S.; Grguric, A.; Mošmondor, M.; Galinac Grbac, T. Ambient orchestration in assisted environment. *Eng. Rev.* **2014**, *34*, 119–129.
25. Grguric, A.; Huljenic, D.; Mosmondor, M. AAL ontology: From design to validation. In Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW), London, UK, 8–12 June 2015; pp. 234–239. [CrossRef]
26. Stav, E.; Walderhaug, S.; Mikalsen, M.; Hanke, S.; Benc, I. Development and evaluation of SOA-based AAL services in real-life environments: A case study and lessons learned. *Int. J. Med. Inform.* **2013**, *82*, e269–e293. [CrossRef]
27. Moyle, W.; Jones, C.; Sung, B.; Bramble, M.; O'Dwyer, S.; Blumenstein, M.; Estivill-Castro, V. What effect does an animal robot called CuDDler have on the engagement and emotional response of older people with dementia? A pilot feasibility study. *Int. J. Soc. Robot.* **2016**, *8*, 145–156. [CrossRef]
28. Cruces, A.; Jerez, A.; Bandera, J.P.; Bandera, A. Socially Assistive Robots in Smart Environments to Attend Elderly People—A Survey. *Appl. Sci.* **2024**, *14*, 5287. [CrossRef]
29. de Carolis, B.; Ferilli, S.; Macchiarulo, N. Ambient Assisted Living and Social Robots: Towards Learning Relations between User's Daily Routines and Mood. In Proceedings of the Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization, New York, NY, USA, 3–7 July 2022; UMAP '22 Adjunct; pp. 123–129. [CrossRef]
30. Marques, G.; Pires, I.M.; Miranda, N.; Pitarma, R. Air Quality Monitoring Using Assistive Robots for Ambient Assisted Living and Enhanced Living Environments through Internet of Things. *Electronics* **2019**, *8*, 1375. [CrossRef]
31. Luperto, M.; Monroy, J.; Moreno, F.A.; Lunardini, F.; Renoux, J.; Krpic, A.; Galindo, C.; Ferrante, S.; Basilico, N.; Gonzalez-Jimenez, J.; et al. Seeking at-home long-term autonomy of assistive mobile robots through the integration with an IoT-based monitoring system. *Robot. Auton. Syst.* **2023**, *161*, 104346. [CrossRef]

32. A Multirobot System in an Assisted Home Environment to Support the Elderly in Their Daily Lives. *Sensors* **2022**, *22*, 7983. [CrossRef]
33. Öztürk, Ö. A Methodology for end user programming of ROS-based service robots using jigsaw metaphor and ontologies. *Intel. Serv. Robot.* **2024**, *17*, 745–757. [CrossRef]
34. Rocha, M.; Favela, J.; Muchaluat-Saade, D. Design and Evaluation of an XML-Based Language for Programming Socially Assistive Robots. *Int. J. Soc. Robot.* **2025**, *28*, 245–258. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.