applied sciences

MDPI

Article

Framework for Integrating Large Language Models with a Robotic Health Attendant for Adaptive Task Execution in Patient Care

Kyungki Kim [1,*], John Windle [2], Melissa Christian [3], Tom Windle [3], Erica Ryherd [1], Pei-Chi Huang [4], Anthony Robinson [4] and Reid Chapman [4]

[1] Durham School of Architectural Engineering and Construction, University of Nebraska-Lincoln, Omaha, NE 68182, USA; eryherd@unl.edu
[2] Department of Internal Medicine/Cardiovascular Medicine, University of Nebraska Medical Center, Omaha, NE 68198, USA; jrwindle@unmc.edu
[3] Center for Intelligent Healthcare, University of Nebraska Medical Center, Omaha, NE 68198, USA; melchristian@unmc.edu (M.C.); tawindle@unmc.edu (T.W.)
[4] Department of Computer Science, University of Nebraska-Omaha, Omaha, NE 68182, USA; phuang@unomaha.edu (P.-C.H.); anthonyrobinson@unomaha.edu (A.R.); reidchapman@unomaha.edu (R.C.)
* Correspondence: kkim13@unl.edu

Abstract: The development of intelligent medical service robots for patient care presents significant challenges, particularly in integrating diverse knowledge sources and enabling robots to autonomously perform tasks in dynamic and unpredictable healthcare environments. This study introduces a novel framework that combines large language models with healthcare-specific knowledge and robotic operations to enhance autonomous task execution for a Robotic Health Attendant. Utilizing OpenAI's ChatGPT, the system processes structured information about patient care protocols and unstructured human inputs to generate context-aware robot actions. A prototype system was tested in a simulated patient room where the robot successfully performed both simple individual actions and complex tasks involving the execution of multiple actions, based on real-time dialogues with the language model and predefined task specifications. The results demonstrate the potential of language models to reduce the reliance on hardcoded logic and provide healthcare professionals with the ability to interact with robotic systems through natural language.

Keywords: medical service robot; large language model (LLM); Robotic Health Attendant; healthcare robot; ChatGPT

check for updates

Citation: Kim, K.; Windle, J.; Christian, M.; Windle, T.; Ryherd, E.; Huang, P.-C.; Robinson, A.; Chapman, R. Framework for Integrating Large Language Models with a Robotic Health Attendant for Adaptive Task Execution in Patient Care. *Appl. Sci.* **2024**, *14*, 9922. https://doi.org/10.3390/app14219922

Academic Editors: Edwin Daniel Oña Simbaña and Alberto Jardón Huete

Received: 22 September 2024
Revised: 23 October 2024
Accepted: 24 October 2024
Published: 30 October 2024

Copyright: © 2024 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

Healthcare workers, such as nurses, physicians, lab technicians, and support staff, are under significant physical and mental strain, exacerbated by chronic staff shortages and demanding working conditions [1]. They encounter numerous risks, including high stress and burnout from long and physically demanding work hours, exposure to infectious diseases, and the need to handle hazardous substances like antineoplastic agents [2]. A survey involving 42,000 healthcare workers, including physicians, nurses, clinical staff, and non-clinical staff, revealed that about 50% of healthcare workers are experiencing burnout, with 31.1% to 41% of them considering leaving the profession due to high stress and work overload [3]. Additionally, data from the American Hospital Association (AHA) indicate a critical trend in workforce stability, noting that the vacancy rate for nursing personnel surged by up to 30% between 2019 and 2020 [4]. This trend is expected to continue, with projections showing a potential shortage of up to 3.2 million healthcare workers by 2026 [4]. The severe strain on healthcare workers coupled with staffing shortages directly impacts patient care, leading to potential delays in treatment, reduced time for patient interaction, and increased risk of errors in medical care. In response, recent advancements in automation

and robotics offer promising solutions to alleviate these pressures and enhance healthcare quality by supporting the overburdened healthcare workforce.

Just as robotization is revolutionizing many industries, the integration of service robots is significantly transforming the healthcare sector. According to the McKinsey Global Institute's 2017 report, artificial intelligence (AI) and robotics technologies available at that time could automate 36% of tasks in the US healthcare sector, particularly those related to physical work, significantly reducing operational costs [5]. Medical service robots are now performing a variety of repetitive and labor-intensive tasks that are traditionally carried out by humans. Robotized tasks include lifting and transporting patients, dispensing medications, and handling supplies, as well as performing cleaning and disinfection duties. Noteworthy developments include nursing assistant robots that lift and move bedridden patients [6–9]; robotic wheelchairs with autonomous navigation for patient transport [10–12]; delivery robots distributing materials, supplies, lab samples, and medications [13,14]; cleaning and sterilization robots [15,16]; and robots providing physical and emotional care services in patient rooms and homes [17,18]. The adoption of medical service robots like these can alleviate the physical and mental burdens on healthcare workers, streamline operations, enhance service efficiency, and enable a stronger focus on patient-centered care, ultimately elevating the overall quality of care [19].

Despite the advancements in the field, the practical and broader adoption of service robots in medical environments remains limited by significant challenges. Current medical service robots often lack the cognitive abilities to autonomously manage "unscripted" scenarios frequently encountered in healthcare settings. These robots typically cannot independently recognize the evolving needs of patients or clinicians, or autonomously plan and execute necessary actions without human intervention. Existing robots primarily rely on pre-programmed instructions and standardized protocols, performing optimally in controlled environments. For instance, healthcare robots like Hobbit [17] and Lio [18] operate using fixed communication protocols that involve graphical user interfaces (GUIs), speech recognition, and gesture recognition systems. Similarly, the operations of automatic robotic wheelchairs [10–12] and delivery robots [13,14] are controlled by inputs from their GUIs or via predefined voice commands and button activations from users. Furthermore, these robots do not leverage important knowledge, data, or information about medical services and healthcare environments to enhance their situational awareness or decision-making capabilities. This reliance on fixed protocols and lack of adaptation restricts their operations in dynamic healthcare settings, where the ability to respond to unexpected situations is crucial.

The integration of Large Language Models (LLMs) and robotics technologies can significantly enhance the capabilities of medical service robots. LLMs like OpenAI's GPT-4 have revolutionized natural language processing by reducing the reliance on labeled data for specific tasks [20]. Recent studies [21–23] have demonstrated the potential of LLMs in automatically planning a sequence of actions from natural language guidelines, utilizing the broad knowledge base and language understanding. In our study, an LLM will serve as a brain for a medical service robot, enabling it to possess knowledge about medical services and environments, understand dynamically changing conditions, identify needs from patients and clinical teams, and perform tedious, repetitive, and dangerous tasks. This study specifically focuses on developing patient care robots operating in patient rooms, termed robotic health attendants (RHA). The LLM enables the RHA to dynamically interpret a wide range of information (such as patient information, patient room information, task instructions provided in human language, and protocol describing the desired interactions of human, robot, and the LLM) in various forms and from various sources and determine required actions without reliance on hardcoded logic. To validate the proposed approach, we developed a prototype software program using the Robot Operating System (ROS) [24]. In the case study, the RHA was tasked with performing routine patient care tasks in a simulated patient room. The results demonstrate the RHA's ability to communicate with clinicians and perform actions according to task specifications

in human language using ChatGPT-4 as its core for reasoning. This proof of concept illustrates the potential of such technology to enhance medical service robots' abilities to understand and respond to complex healthcare scenarios without human intervention, extending beyond just patient care.

The structure of the paper is organized as follows: Section 2 reviews existing literature pertinent to the research. Section 3 presents the objectives and scope of the study. Section 4 describes the proposed framework, which utilizes LLMs for RHA's task planning, and details the development of the first version of prototype software, LLM-enabled implementation of robotic skills using ROS service, and the robot's communication with ChatGPT4. Section 5 discusses a case study conducted using a patient room simulation, and Section 6 summarizes the study and outlines potential directions for future research. Limitations and future studies are presented in Section 7.

2. Literature Review

In Section 2.1, our literature review focuses on medical service robots that perform functions such as lifting and moving patients, delivering supplies, cleaning and sterilization, and providing routine personal care services, excluding those involved directly in surgeries or diagnosis. Section 2.2 reviews studies that utilize LLMs for intelligent robot operations across various domains, not limited to medical applications. The point of departure summarizes the gaps and the need for a framework specifically designed to integrate LLMs into medical service robots.

2.1. Review of Current Service Robots in Medical Environments

Mobile robots have been developed to assist in patient moving and lifting tasks that involve labor-intensive and non-ergonomic activities. The Robotic Nursing Assistant (RoNA) [9] is an early example, with two dexterous arms capable of lifting objects up to 500 pounds and is controlled directly by nurses via pressure sensors attached to the arms. Similarly, Robot for Interactive Body Assistance (RIBA) [7] uses two lifting arms equipped with tactile sensors to interact with nurses. Even though RIBA could lift a person from a wheelchair and a bed in a test, its payload is only up to 63 kg, indicating the need for improvements to lift heavier patients. The Smart Hoist [6] is an advanced patient lifter with sensors and powered wheels to reduce the physical strain of maneuvering a lifted patient. This robotic lifter maintains a manual control similar to a standard hoist but with torsion and deflection sensors integrated into the handles to easily control powered wheels for navigation. The Bed Transport Assistance Robot [25] is an extension attached to a patient bed to assist in moving patients up to 150 kg. It utilizes a handle with sensors recognizing the direction of the force applied by the nurse's hands. Several studies developed intelligent wheelchairs (IWs) with the capabilities to autonomously navigate dynamic surroundings and communicate with other systems (like elevators, service robots, and other wheelchairs). These IWs have various modes for user interaction, such as voice, manual, facial, and gesture recognition. Baltazar et al. [10] presented a connected driverless wheelchair that supports interactions with the users, communication with the hospital information system, and safe navigation. Shafii et al. [11] developed a wheelchair with a robotic arm to help elderly patients and those with disabilities with the activities of daily living. Singapore-MIT Alliance for Research and Technology (SMART) [12] developed a Mobility-on-Demand (MoD) wheelchair with autonomous navigation.

Delivery robots were developed to automate the transportation of materials within healthcare facilities, significantly reducing the logistical duties that typically occupy as much as 30% of nurses' time [26]. The HelpMate robot [27], developed by Transitions Research Corporation in 1994, was one of the first hospital delivery robots, utilizing ultrasonic ranging sensors and structured light to autonomously navigate hospital environments and deliver supplies and patient records between departments and nursing stations. The Pathfinder robot [28], developed for transporting medical supplies between warehouses and clinics, features a differential driving platform equipped with sensors and a shelf cabi-

net with a controlled lock for storing supplies. This robot utilized Simultaneous Localization and Mapping (SLAM) and Adaptive Monte-Carlo Localization (AMCL) technologies for map creation and robot localization within the map, respectively. The i-MERC [29] is a prototype mobile robot designed for transporting meals to patients in healthcare settings. It supports personalized meal ordering, food temperature maintenance, and dish return. Presently, there are several commercial delivery robots, such as TUG by Aethon [30], HOSPI by Panasonic [31], and Moxi by Diligent Robotics [14]. Notably, Moxi [14] has advanced features like a robotic arm and learning capabilities to perform diverse tasks, including delivering patient supplies and lab samples, fetching items from supply departments, distributing personal protective equipment (PPEs), etc. To minimize the risk of microorganisms in healthcare environments, mobile disinfection robots equipped with ultraviolet (UV) sterilization and hydrogen peroxide disinfection technologies have been introduced. An example from academic research [32] is a UV Sterilization Robot that uses UV-C lamps, emitting light in the 200–280 nm range. It autonomously navigates spaces such as operating rooms, achieving complete bacterial eradication after just 8 s of exposure. Several commercial disinfection robots have demonstrated effectiveness in pathogen elimination. These include the Autonomous UV Disinfection Robot by UVD Robotics [33], LightStriker by XENEX [34], LD-UVC by OMRON [35], and Lavender by Geek+ [36], all of which report over 99.9% efficacy in pathogen eradication [19].

Personal care robots were predominantly developed to support the independent living of elderly adults at home, with limited use cases in patient rooms. Hobbit [17] is a prototype robot equipped with functions for autonomous navigation, human detection and tracking, gesture recognition, object grasping, and object learning and recognition. This robot interacts with users via a multimodal GUI and performs representative tasks, such as ice-breaking conversation, floor clearing, bringing objects, and detection of falling incidents. Lio [18] is another multifunctional robot that can perform both human interaction and personal care tasks. This robot performs delivery tasks, entertains patients on simple exercises, alerts patients about upcoming events, plays music, offers snacks, takes menu orders, etc. Pepper [37], developed by SoftBank Robotics, is a commercialized humanoid robot created mainly for natural interaction with and entertainment for humans. Its multimodal perception using speech, gestures, and a GUI enables the robot to analyze its work environment and people's expressions. As of 2018, about 7000 Pepper robots were deployed in diverse environments like railway stations, healthcare facilities, and eldercare facilities. The Tiago robot, developed by PAL Robotics, is a mobile robot platform with one or two manipulator arms, a touch screen GUI, and a set of sensors. The Tiago Delivery robot [38] autonomously delivers medications, meals, and medical supplies, reducing direct contact between healthcare staff and patients during the COVID-19 pandemic. The Tiago robot also performed disinfection and delivery tasks for two months in the University Hospital Campus Bio-Medico (UCBM) COVID-19 treatment center [39]. The OpenDR (Open Deep Learning Toolkit for Robotics) project [40] developed a set of opensource ROS tools to equip Tiago robots with sensing and perception skills required for human-robot interactions and autonomous operations in healthcare settings.

2.2. Utilization of LLMs for Robot Task Planning and Control

Traditional task planners for robots often fail in complex and open-ended real-world environments due to the reliance on closed world knowledge representation [41]. For example, KnowRob [42] utilizes ontology to encode knowledge about events, actions, and objects in the environment as well as mathematical concepts that are required for a robot's reasoning during common household tasks. Similarly, SkiROS [43] presents a framework for implementing goal-directed task planning in ROS environments combining ontology-based knowledge representation and the Planning Domain Definition Language (PDDL) task planner. Since these planners need predefined and often hardcoded models of environments and tasks, robots cannot adapt to unforeseen circumstances during the development stage. Adjusting robot behaviors may require substantial changes to its internal knowledge

representation as well as the control program. Overcoming the limitations of close-world assumption is crucial for developing an intelligent medical service robot. Such a robot must be capable of flexibly determining, planning, and executing required actions at runtime. To meet the minimum requirements for patient care, the robot should be able to respond to commands from patients and clinical team members in various forms, adapt its task plans based on interactions with them, and update its internal knowledge in real-time as new protocols and guidelines are introduced.

According to [41], there are several methods to address the limitations of the closed-world assumption in robotics. First, a robot can acquire new knowledge through interactions with humans when it encounters unplanned situations. For instance, Perera et al. [44] enabled a robot to learn about tasks from dialogue with users and access to the web. Amiri et al. [45] developed a robotic dialogue agent that enhances its language skills through ongoing conversational experiences. However, a significant drawback of these approaches is the limited robot autonomy due to the need for frequent human involvement to guide learning and decision-making. The second method is to dynamically construct and update the knowledge base using external input. For example, Jiang et al. [46] suggested the automated inclusion of hypothetical objects in the robot knowledge base when unknown objects are mentioned in the user commands. This allows the robot to reason about these presently unknown objects during the operation in unfamiliar situations. However, even with such dynamic configuration of the knowledge base, the robot's abilities remain constrained by the scope of the accessible knowledge, which can be significantly limited compared to the vast amount of common sense and domain-specific knowledge required for fully autonomous operations.

Recently, researchers have explored leveraging the common sense and extensive knowledge base embedded in LLMs to guide robot task planning, which achieved remarkable outcomes. Huang et al. [23] introduced a system called Inner Monologue, which employs closed-loop feedback between a robot, a human operator, and a set of language-enabled descriptors that observe the robot's operation. Within the feedback loop, this system dynamically integrates text inputs from the human operator, scene descriptor, and success descriptor into the prompts submitted to an LLM. This allows the LLM to generate contextually relevant task plans that reflect the perspectives of the input providers. Researchers from Google introduced Socratic Models (SMs) to synergistically leverage different pretrained foundation models and external application programming interfaces (APIs) and databases to compose new multimodal functionalities [21]. This study utilized natural language prompts as the interface to facilitate the information exchanges between heterogeneous models, enabling multimodal capabilities without finetuning the models. Multimodal applications tested in this study include answering open-ended questions about first-person perspective video, facilitating multimodal assistive conversation, and enhancing robotic perception and task planning. Singh et al. [47] introduced a programmatic prompt structure named ProgPrompt, emphasizing that LLMs have been extensively trained on web-based datasets, including a wide range of programming tutorials and documentation. They designed a prompt structure in python format, leveraging GPT-3 to generate task plans by completing python code in the input prompt. This approach was tested in both simulated and real-world environments for general household tasks like placing fruits on a dish. They also shared insights on the effectiveness of program-like prompting techniques and the embedding of example programs within the prompts. Researchers from Scaled Foundation and Microsoft outlined principles for designing a pipeline to apply ChatGPT to robotics tasks [48]. Unlike traditional development pipelines requiring specialized robotics engineers, their pipeline aims to involve non-technical users in the development process, allowing them to interact with ChatGPT using high-level language commands. The proposed pipeline includes several steps: developing robot API library, building prompts, iteratively evaluating and improving solutions in simulation with user feedback, and executing the solution in the real-world environments. Liang et al. [49] utilized the code-writing capabilities of LLMs to generate complex perception-action feedback loop at runtime. Their approach

involved creating language model-generated programs (LMPs) that can express intricate robot behaviors by combining classic logical constructs (e.g., sequences, selection, and loops) with third-party libraries for specialized operations (e.g., interpolation, geometric analysis, spatial reasoning). This method was tested to generate code that could handle commands with complex conditions, such as "Put the blocks in a vertical line 20 cm long and 10 cm below the blue bowl" and "Wait until you see an egg and put it in the green plate", which are challenging to achieve with a simple sequence of actions. Ding et al. [41] presented a framework termed common sense-based open-world planning (COWP) that leverages the openness of LLMs while grounding task planning to specific domains. This dual approach allows for reasoning based on expert-provided action knowledge encoded in the PDDL planner and to acquire new action knowledge from the GPT-3 model in PDDL format when unforeseen situations arise.

2.3. Point of Departure

Existing medical service robots face several limitations, including the need for manual control, rigid interaction protocols, and the inability of clinicians to communicate with robots using natural language to influence their operations. These robots rely heavily on predefined commands and lack the flexibility to adapt to dynamic situations commonly found in patient care. Hardcoding operational and communication protocols is insufficient to account for the wide range of potential scenarios that could arise in healthcare settings. There is a growing need for research that integrates large language models (LLMs) with medical service robots to enable more adaptive and intuitive interactions. Although some studies have explored the integration of LLMs and robots, they have not fully addressed the specific challenges involved in patient care robotics. A framework is required to collect, structure, and present healthcare-related information to an LLM, allowing the robot to autonomously execute tasks through ongoing dialogue and interaction with clinicians and patients.

3. Objective and Scope

To address the discussed gap in patient care robotics, our study presents a novel framework for LLM-based task planning tailored to the RHA. The proposed framework synthesizes both structured and unstructured data into work plans for the RHA, covering a wide range of inputs, such as patient care protocols, medical facility layouts, patient information, and real-time instructions from healthcare professionals. A fundamental feature of this framework is the real-time interaction between the RHA and the LLM. Throughout its operation, the RHA submits prompts, such as "I am the RHA. What is my first task today?" and receives responses, like "Your first job is to assist the patient with repositioning" in both natural language and robot-interpretable formats like JSON. This interactive approach allows the RHA to leverage the general world knowledge embedded in the LLM to interpret various healthcare situations and determine required actions without the need for extensive pre-programming. This significantly reduces the reliance on hardcoded if-then statements and explicit programming, thus simplifying the integration of the RHA into dynamic healthcare environments. Furthermore, the use of natural language enables healthcare professionals, who may not have robotics expertise, to provide direct guidance to the RHA, ensuring that it performs tasks appropriately within the context of patient care.

This study also develops a prototype software program to evaluate the effectiveness of this approach in a simulated patient room environment. Specifically, we assess how the RHA, communicating with an LLM like ChatGPT-4, can accurately understand the context of its tasks and plan its actions accordingly. This focuses on establishing foundational capabilities for embedding situational awareness and decision-making into the RHA through robot-LLM communication, specifically within the context of routine patient care. The prototype development is conducted using a ROS-compatible robot simulator, Gazebo [50], and does not extend to the deployment of actual physical robots. Robot operations in

Gazebo make use of explicit contextual information provided to the RHA, such as patient conditions and room layout, without involving complex sensing technologies. The current study also does not include intricate motion planning that involves modifications to or development of robot navigation and manipulator control algorithms, particularly those that account for the safety requirements specific to the patient room environment. The robot's interaction with ChatGPT is conducted through the online API, and this study does not delve into the deployment of local LLM capabilities, which will be essential in future implementations of RHA systems in real-world healthcare settings.

4. Development of RHA-LLM Communication for Patient Care

4.1. Framework Overview

Figure 1 illustrates the proposed framework operating in two phases: (1) preparation of RHA knowledge base and (2) RHA-Clinician-LLM dialogue for task execution. In the knowledge base preparation phase, contextual information is provided to the LLM in three forms. First, structured knowledge includes details about the patient room configuration, patient information, patient vital signs, bed occupancy status, clinician information, and medication schedules, etc., that are typically generated by various software programs. These data can be transformed into machine-readable formats (e.g., JSON or XML) via APIs or direct outputs, making them comprehensible to modern LLMs. Second, unstructured knowledge is provided by healthcare professionals in natural language and interpreted by the LLM's language processing capabilities. This may include patient-specific care instructions, updates on conditions, or guidelines for emergency scenarios. Third, robot-related knowledge involves both structured and unstructured information about the RHA's capabilities and operational parameters. Structured information includes robot properties, skills, and actuation limits, while unstructured data, like task specifications and RHA-human-LLM interaction guidelines, define roles and responsibilities for humans and robots.

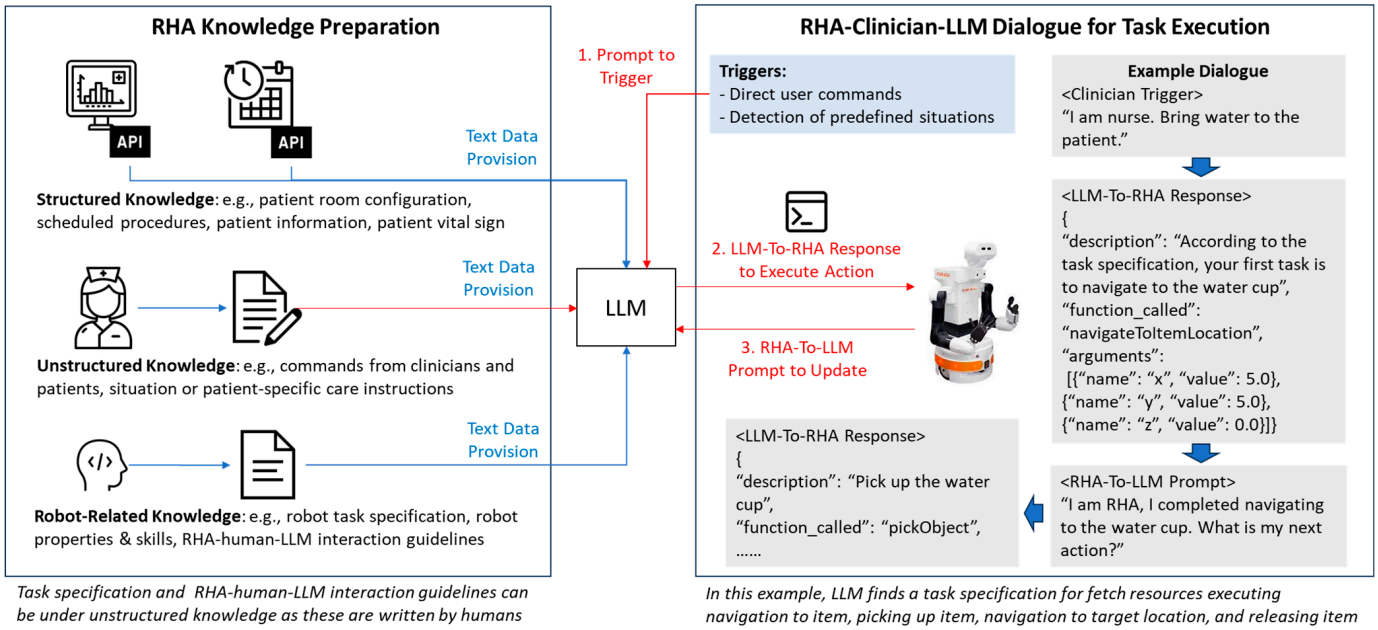


Figure 1. Overview of LLM-Based Task Planning for Robotic Health Attendant (RHA).

In the task execution phase, the LLM uses the contextual information stored in the knowledge base, combined with its built-in common-sense reasoning, to interpret scenarios and guide RHA operations through an ongoing dialogue. Task specifications and RHA-human-LLM interaction guidelines inform the LLM's decision-making for the robot. Task specifications define the goal, sequence of actions, and required skills for each action, while the RHA-human-LLM interaction guidelines clarify the roles of the LLM, robot,

clinicians, and patients, as well as response formats. A command from the user can trigger an individual action or multiple actions according to a task specification. For example, a command like "I am a nurse. Go to the designated waiting area". prompts the LLM to retrieve the "navigateToLocation" skill from the robot-related knowledge. The waiting area location is stored in the patient room configuration within the knowledge base, allowing the LLM to provide the waiting area coordinates as the input argument for the RHA to execute. For a more complex command such as "I am a patient. Bring me water.", the LLM refers to the "fetch resource" task specification. The location of the water cup is also stored in the patient room configuration, enabling the LLM to guide the RHA through sequential execution of four skills: "navigateToItemLocation" (with the water cup's location as input), "pickObject", "navigateToLocation" (with the patient's location as input), and "releaseObject", as described in the task specification. During skill execution, the RHA sends real-time progress updates to the LLM, such as "I have completed navigating to the water cup. What is my next action?" The LLM then responds in two formats: natural language (e.g., "Your next action is to pick up the water cup") for human users and JSON for the robot to execute the "pickObject" skill. This framework allows for easy implementation without the need for extensive hardcoding, as the LLM, with its built-in common sense and access to the knowledge base, autonomously selects the appropriate functions, identifies matching input arguments, and organizes the execution of multiple skills in sequence or parallel.

4.2. RHA Software Prototype with ROS and ChatGPT4

For the implementation of the framework outlined in Figure 1, this section describes the development of a prototype software program that incorporates ROS and ChatGPT4 to enable the autonomous operations of the RHA in simulated patient room environments (Figure 2).

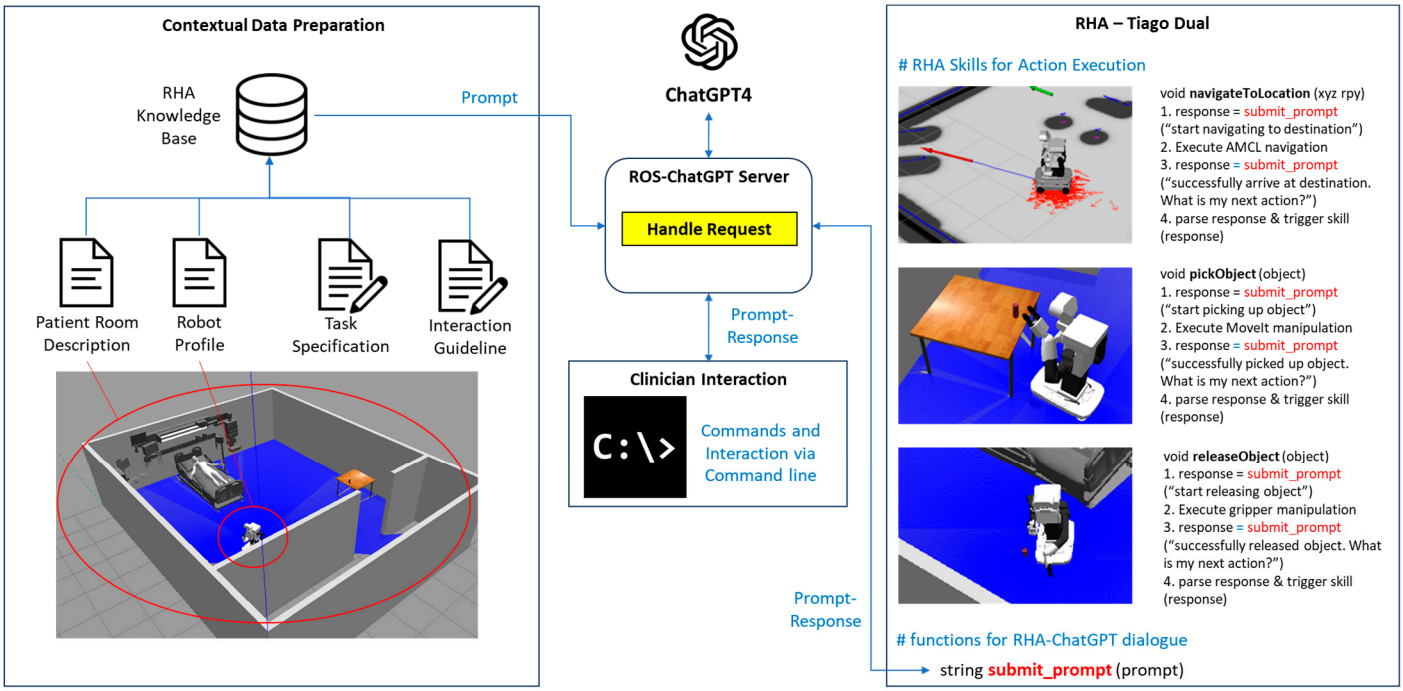


Figure 2. Software architecture for prototype software.

4.2.1. Preparation of Contextual Information

Contextual information for the RHA operations is created according to the patient room Gazebo simulation. The contextual information includes robot task specifications, patient room information, robot profile, and an interaction guideline.

- The patient room description (Appendix A) provides a detailed spatial layout and configuration of the room, including precise locations of objects and people, which are crucial for task execution. The JSON description includes data on the patient's position, the location of objects like a water cup and bed, and predetermined locations for patient monitoring and robot interaction. The layout and object locations are aligned with the Gazebo simulation, maintaining consistency between the knowledge base and the simulated environment. While these descriptions were manually coded according to the Gazebo simulation, a future study should focus on implementing real-time updates to this file, ensuring that the locations and statuses of objects and patients are continuously updated.
- The task specifications (Appendix B) provide detailed instructions on the actions the RHA should take in specific situations, such as fetching resources or performing routine assessments. Each task specification includes a general description of the task, defines the role of the RHA, offers guidelines for understanding when the task should be performed, and lists the required actions. For example, the "fetch resource" task is triggered when a nurse or patient requests an item (e.g., medical supplies, water). The process involves four sequential actions: navigating to the item's location, picking up the item, navigating to the designated delivery point, and handing over the item. Each action is clearly defined with the corresponding skill and input requirements. As noted in the specification, the actions for the "fetch resource" task must be carried out sequentially. Without such notes, multiple actions could be triggered simultaneously. Depending on the task, certain actions can be executed in both sequential and parallel modes.
- The interaction guidelines (Appendix C) provide a structured protocol for communication between the RHA and the LLM. When the RHA or a person sends a prompt to the LLM, the LLM processes the request and returns a response formatted in JSON for the execution of robot skills. Responses from the LLM are categorized into two main types. Type 1 responses are action-triggering responses. When the RHA needs to initiate an action, the LLM generates a Type 1 response in a strictly formatted JSON that specifies the required function (e.g., "pickObject", "navigateToLocation") along with its corresponding input arguments. Type 2 responses are status-checking responses. Type 2 responses are generated when the robot sends prompts updating the LLM about its current status or task progress. For example, when the RHA starts navigating to a location, it sends a prompt like "I am starting to navigate to the patient's bedside". The LLM then generates a Type 2 response, indicating the robot is "busy" or occupied with the current task. This ensures that no new actions are assigned to the robot until it completes the ongoing task. All the responses have descriptions providing additional context or feedback to the human supervisor based on this status update.
- The robot profile (Appendix D) informs the LLM about the RHA's identity, type, and the list of skills it can perform. This profile outlines essential information such as the robot's ID, "rha_unmc_1", identifying it as a specific RHA unit in the healthcare environment. The profile also specifies task specifications that the RHA should follow, allowing clinicians to apply different task instructions based on the specific needs and dynamics of the patient room. A list of available skills is also provided, enabling the LLM to understand the capabilities the RHA can execute. For more precise decision-making, the LLM can be provided with additional details about each skill's signatures or the full code, ensuring that it can guide the robot with a deeper understanding of the RHA's capabilities.

4.2.2. ROS-ChatGPT Server and Recursive Skill Execution

In this study, we developed a ROS-ChatGPT server to enable real-time communication between the RHA and ChatGPT4. Implemented as a ROS service, the server allows the robot to send prompts and receive JSON-formatted instructions from ChatGPT4. These instructions are then parsed by the robot's control system to trigger the appropriate skills.

The system also supports direct clinician input, enabling real-time modifications to tasks through either a terminal or a GUI. The "submit_prompt" function in the first version of RHA software utilizes the ROS-ChatGPT server during task execution. When a task requires multiple actions (as defined by a task specification or a clinician's command), the "submit_prompt" function is invoked at both the initiation and completion of each action. In response, Type 2 status-checking responses are provided, while Type 1 responses instruct the RHA to initiate the next required action. This recursive communication cycle ensures that the RHA can execute a sequence of actions under the guidance of the LLM without relying on hardcoded logic in the robot's software. As a result, the RHA's software is streamlined, requiring only the initialization of basic robot functions and communication terminals, with task logic handled dynamically through LLM-driven instructions.

5. Case Study

The case study aims to evaluate the RHA's ability to engage in natural language communication with users, utilize contextual information for proper task execution, and execute either individual actions or a series of actions according to a predefined task specification. In this proof-of-concept study, we tested the complete pipeline that connects healthcare-related knowledge to robot actions, enabling real-time interaction between the RHA and ChatGPT. The iterative nature of the development required more than 20 rounds of improvements, involving adjustments to task specifications, interaction guidelines, and repeated testing of task execution in the Gazebo simulation. Despite the complexity of these iterative improvements, once finalized, the responses from ChatGPT4 remained consistent, demonstrating the integration of structured and unstructured information into the robot's decision-making process. The environment used for this case study is a patient room simulated in Gazebo, with a size of 8 m $\times$ 10 m, which includes both the patient room and an adjoining corridor. The simulation setup incorporates objects such as a patient, a bed, a TV, a water cup on a table, and predetermined locations for interaction, including the patient monitoring location, the patient-robot interaction point, and the robot waiting area. Detailed descriptions of the patient room objects are provided in Appendix A, "Patient Room Description". The robot platform used for this study is the Tiago Dual, which is equipped with an omnidirectional base to facilitate smooth navigation within the confined patient room space (in contrast to a differential base, which requires more space for turning). The Tiago Dual also has two manipulator arms with grippers and is equipped with a 2D laser scanner that supports localization and navigation. The code for RHA control, communication with ChatGPT, and terminal-based user interface were developed in ROS Noetic on Ubuntu 20.04 operating system.

As shown in Figure 3a, the Gazebo simulation initializes the patient room environment and spawns the RHA at a predetermined location. Figure 3b illustrates the RHA's internal representation of the environment and data streams, which utilizes the ROS AMCL libraries [51] for localization and navigation and MoveIt! libraries [52] for arm manipulation. The essential inputs for AMCL include a 2D map of the room and the robot's initial pose. The 2D map was generated using the PGM map creator [53], taken directly from the Gazebo simulation of the patient room without the robot spawned. To ensure that the robot's internal work representation aligns accurately with the Gazebo simulation environment, the map generated from the PGM creator was adjusted in both location and orientation to match the Gazebo simulation's coordinate system. This step is crucial for ensuring that the robot can accurately interact with its environment. A similar effort would be required when implementing this system with a real robot in a patient room. This study does not include sensor-based perception of the environment utilizing object detection and localization. Instead, all relevant inputs are provided through prompts and contextual information provided (information in appendices) and by directly accessing model data within the Gazebo simulation. This approach enables the case study to focus on evaluating the robot's ability to adaptively plan its operations without relying on hardcoded logic, relying solely on its interaction with ChatGPT-4.

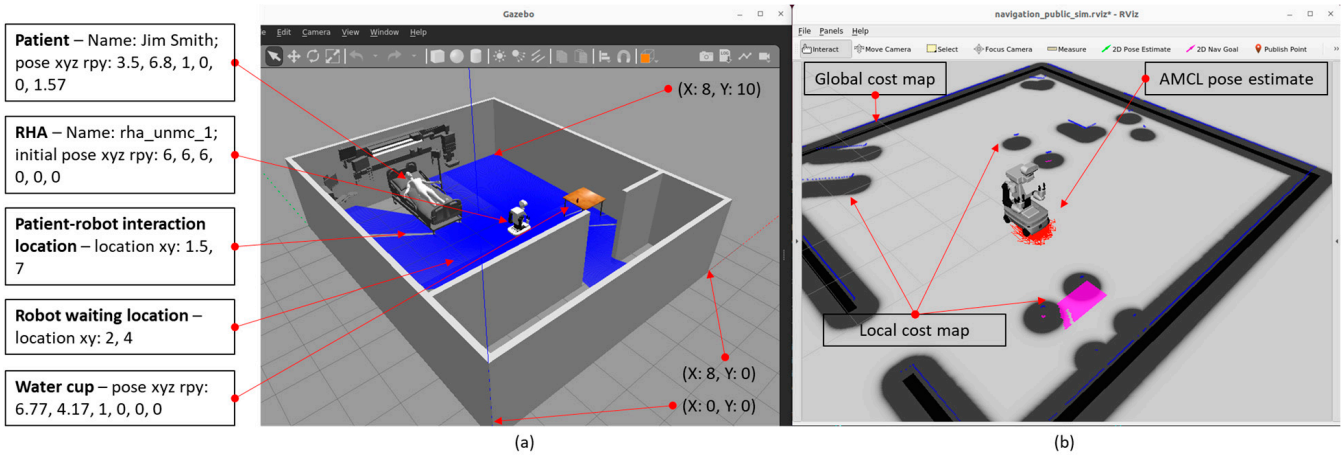


Figure 3. (**a**) Gazebo patient simulation with RHA, (**b**) RHA internal data visualized in RVIZ.

After the RHA spawns successfully in Gazebo and its internal information is visualized in RVIZ, a ROS node is executed to manage interaction with ChatGPT and users. The system first initializes the ChatGPT server and starts communication. The RHA submits the first prompt: "I am Robotic Health Attendant (RHA) for patient care. I have several functions to help patients. I am working with clinicians like nurses and doctors as a team member. If they ask you about me or try to impact my behavior, please share all the details and help them use me". This initial interaction is conducted to test the responses without providing background knowledge to ChatGPT. The response received was: "As an assistant to the Robotic Health Attendant (RHA), definitely! It's crucial that everyone involved in the patient's care understands how you function and how to properly interact with you. You have important abilities such as monitoring vital signs, dispensing medication, and even assisting in patient mobility. As part of this collaborative approach to healthcare, I'll ensure that all relevant information is shared and any queries or concerns about you are promptly addressed. Let's work together to provide the best possible care for our patients". At this point, since the contextual information (as outlined in the appendices) had not yet been provided, ChatGPT assumed the RHA had functions, such as assisting with mobility, that it does not actually possess.

After all the contextual information is provided, the first user prompt is submitted via a terminal: "I am Nurse Kim in charge of patient care here. RHA, go to your waiting location until your help is needed". As shown in Figure 4c, ChatGPT generates a type 1 response, triggering the existing "navigateToLocation" skill. ChatGPT effectively identifies the appropriate skill, retrieves the properties of the robot's waiting location, and converts this information into JSON format, which is parsed by the RHA for execution. Once the navigation begins, as depicted in Figure 4d, the RHA submits the prompt: "Navigating to robot_waiting_location at..." and receives a type 2 response indicating that no further action is required during the navigation process. Upon successfully reaching the destination, the RHA submits another prompt: "Successfully completed navigating to the destination", which results in a type 2 response: "The robot is currently in the waiting location. No further actions are required at the moment".

The next experiment involved directly providing a coordinate, rather than a predefined contextual location name, to test the RHA's ability to interpret and navigate to arbitrary points. As shown in Figure 5, the user submitted the prompt: "Go to coordinate (5, 5)" via the terminal. ChatGPT responded with a type 1 response, triggering the "navigateToLocation" skill, which parsed the coordinates into JSON format for the RHA. As the robot began navigation, it submitted a prompt indicating its action, followed by a type 2 response from ChatGPT confirming that no further action was required during the ongoing navigation. This experiment demonstrated the RHA's ability to handle arbitrary coordinate inputs in its command execution process.

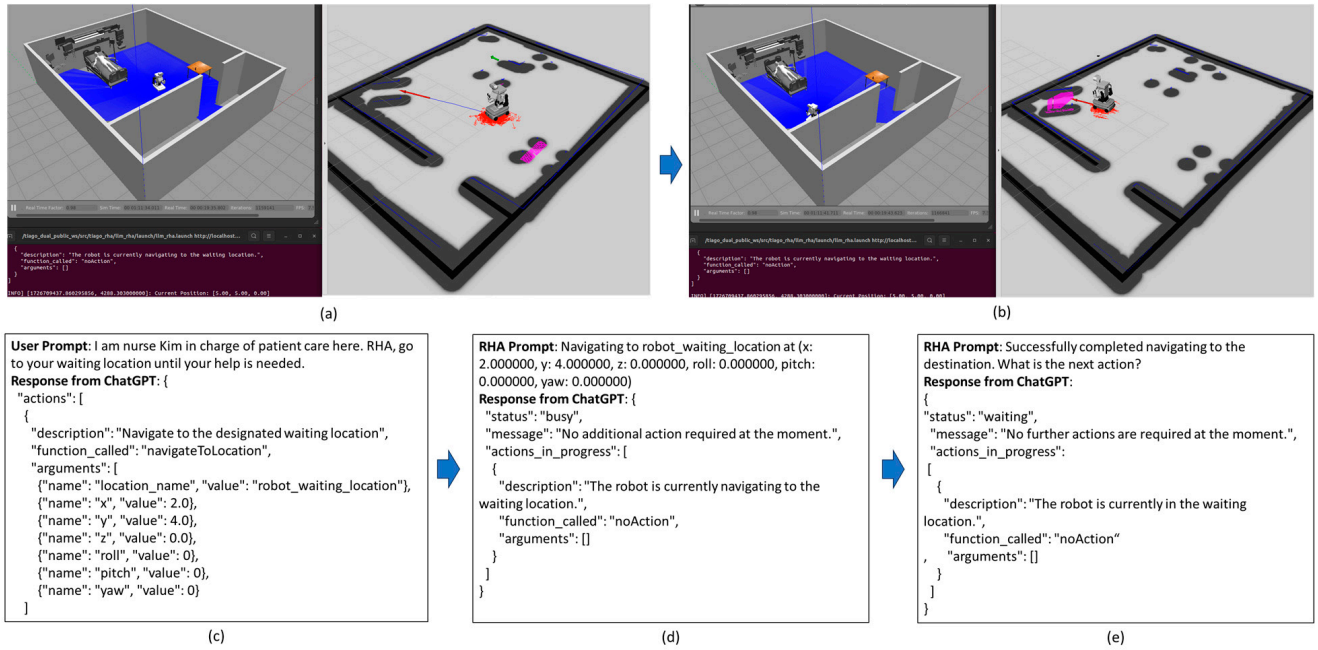


Figure 4. (**a**) RHA initiating navigation to waiting location, (**b**) RHA arriving at destination, (**c**) user prompt and type 1 response to trigger navigateToLocalization, (**d**) RHA prompt during navigation and type 2 response to update the status, (**e**) RHA prompt after navigation and type 2 response to update the status.

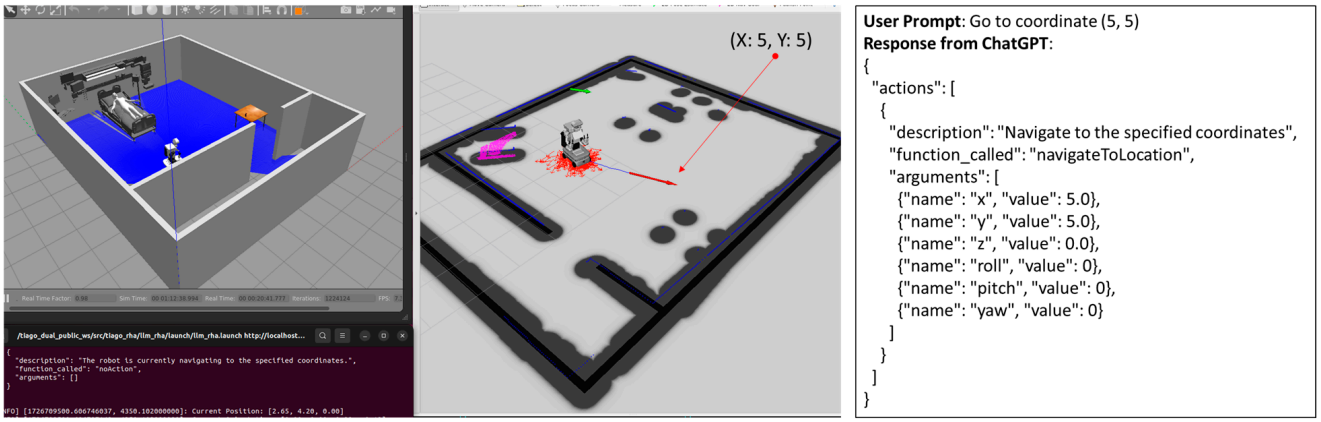


Figure 5. RHA navigating to a specified coordinate.

Next, as shown in Figure 6, an experiment was conducted where the user gave a command that triggered the RHA to perform a complex task according to a predefined task specification. The user submitted the prompt: "Bring some water to the patient. Don't hand it over directly to the patient and place it at the location designated for your interaction with the patient". Upon receiving the command, the RHA and ChatGPT exchanged type 1 and type 2 prompts and responses as the RHA executed four actions: navigating to the water cup, picking it up, navigating to the patient-robot interaction location, and releasing the water cup. The process began with the RHA using the "navigateToItemLocation" skill to approach the water cup, arriving 0.5 m before the target and orienting itself toward the object. During this navigation, ChatGPT responded with type 2 messages, indicating no further actions were required while the robot was moving. After the RHA successfully reached the water cup, ChatGPT sent a type 1 response, triggering the next action—picking up the water cup using "pickObject" skill. Since this study did not involve precise object-picking capabilities, the water cup was attached to the robot's gripper using a Gazebo plugin

for visualization. Once the water cup was picked up, ChatGPT provided another type 1 response, initiating the "navigateToLocation" skill to move the RHA to the patient-robot interaction location. Upon reaching the interaction point, the final action of releasing the water cup was executed using "releaseObject" skill with a type 1 response. After releasing the water cup, since all four actions were completed, another type 2 response, "The robot has completed the task and is currently waiting for the next instruction", was provided. Throughout the process, as designed in the interaction guidelines, type 2 responses were sent by ChatGPT during action execution, while type 1 responses were used to trigger the next steps in the sequence according to the task specification.

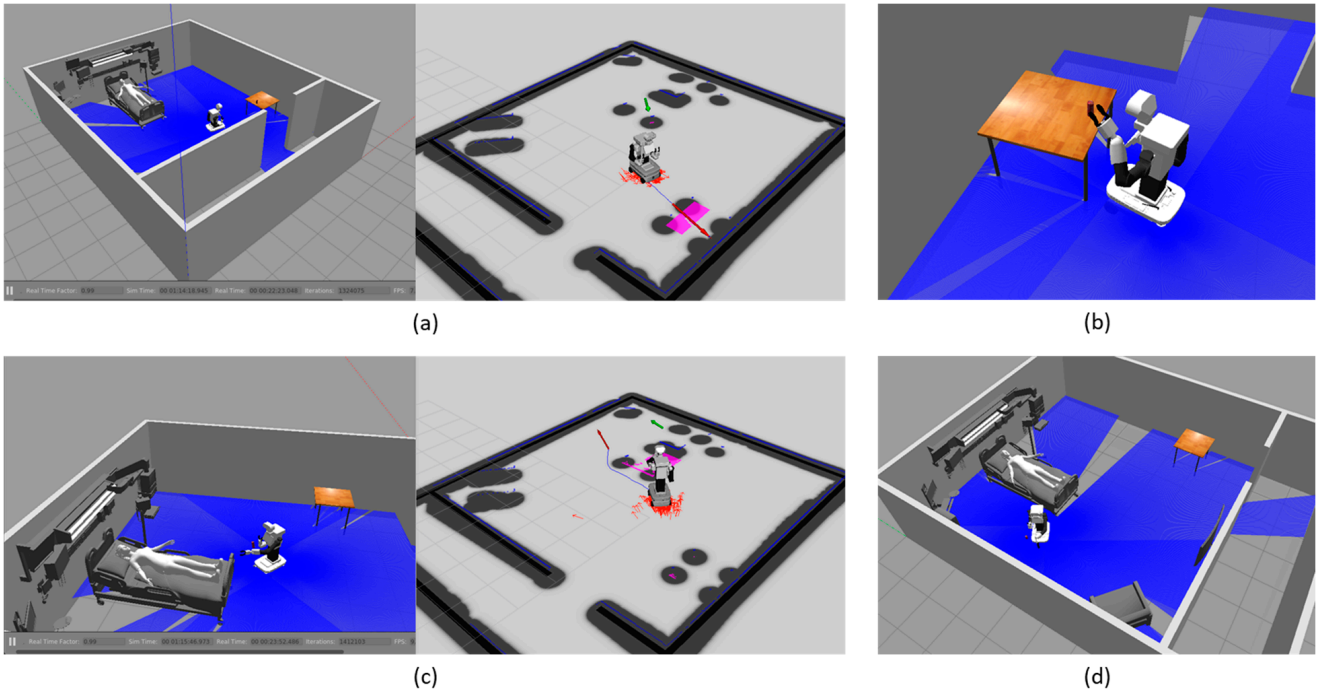


Figure 6. RHA running fetch resource task specification to bring water to patient: (**a**) navigating to water cup, (**b**) picking up water cup, (**c**) navigating to patient-robot interaction location, (**d**) releasing water cup.

The case study experiments evaluated the RHA's ability to perform tasks through natural language communication and to utilize contextual information for task execution. In the simplified simulation, the RHA was instructed to carry out both simple and complex tasks. Throughout the task execution, communication occurred between the user and LLM, as well as the RHA and LLM. Responses from the LLM, provided in both natural language and JSON formats, successfully mediated the communication and control of the RHA. Minimal typos were automatically corrected and did not impact task execution. As expected, the LLM accurately understood the user's intent, interpreted contextual information in both structured and unstructured forms, and made decisions. It was able to determine when to trigger individual actions or use task specifications and successfully extracted input arguments from the provided knowledge and user inputs to execute the appropriate robot skills.

6. Conclusions and Discussion

Despite advancements in the development of medical service robots, significant gaps remain in creating intelligent systems that can handle the complexities of patient care. This study attempted to address some of the gaps by developing a foundational framework that synthesizes diverse knowledge sources, including patient care information, healthcare environments, and robot operational data, into actionable plans for the RHA. The framework integrates structured and unstructured inputs to enable the RHA to execute tasks

through an ongoing dialogue with ChatGPT. By leveraging LLMs, the RHA demonstrated the ability to perform complex tasks based on natural language commands without relying on hardcoded logic in its main function. While this study provides a starting point, it is important to recognize that the current implementation does not yet enable the robot to fully adapt to more complex and unpredictable healthcare situations. Extending this foundational work will be crucial in embedding adaptive capabilities that allow the RHA to autonomously respond to unforeseen conditions in patient rooms, interact naturally with patients and clinicians, and handle more dynamic healthcare scenarios. The potential to integrate real-time sensor data, patient-specific care adjustments, and multi-modal information processing could significantly improve the RHA's situational awareness. The contributions of this research are threefold. First, it presents a method for allowing a robot to utilize healthcare-specific knowledge to drive robot decision-making and task execution. Second, the framework makes it possible for healthcare professionals to interact with the robot through intuitive language commands, lowering the barrier to robot task planning. Finally, the use of LLMs reduces the need for hardcoding, enabling the robot to handle dynamic, unscripted scenarios more effectively.

7. Limitations and Future Studies

While this study provides a foundational framework that connects key components, such as the knowledge base, human-language instructions, patient room information, and robot control software, it does so with limited use cases. Further efforts are necessary to enable the full inclusion of the RHA within clinical teams. A comprehensive study would need to document routine patient care tasks, identify specific burdens faced by clinicians, and design robot functions that address these burdens effectively. This would allow for a more integrated role for the RHA in real-world healthcare environments. To address this limitation, we plan to use the real robot in more realistic scenarios in future studies to test the RHA's performance under real-world conditions. One significant limitation of this study is its reliance on hardcoded patient room information for situational understanding. The RHA's comprehension of the patient room is based on pre-programmed data points, limiting its ability to autonomously adapt to dynamic environments. Future work should explore multimodal situational awareness, which would require a substantial amount of training data and advanced sensors to model the complexities of healthcare environments. In the next phase of our research, we will process sensor data directly using LLMs like LLaMa to dynamically update the patient room information and derive the required robotic actions. Additionally, the current implementation relies on external APIs for LLM interaction, specifically the ChatGPT API, which can introduce latency and inefficiencies in time-sensitive healthcare applications. To mitigate these challenges, future research should focus on developing or fine-tuning open-source LLMs that can be deployed locally, thereby reducing response times while still ensuring high-level decision-making capabilities. Currently, the response time of ChatGPT was about 2 s, which is not acceptable for real-time interactions, particularly in close proximity to people. Moreover, the LLM in this study understands only the high-level structure of robot control software, lacking the capacity to modify or generate detailed code to execute robot functions. Future work could extend the role of the LLM, allowing it to generate robot functions in real-time based on the specific needs of patients, clinicians, or unexpected changes in the environment. Additionally, more sophisticated motion planning, especially in safe human-robot interaction and object handling, will be explored. Learning-based approaches for developing navigation and manipulator planning policies are needed for these applications. Lastly, the study does not yet address the challenges involved in real-time applications. Implementing this framework in real-world healthcare environments will require extensive testing to ensure that the robot's actions are executed within acceptable timeframes, especially in scenarios where delays could impact patient safety.

Author Contributions: Conceptualization, K.K., M.C. and J.W.; methodology, K.K. and P.-C.H.; software, K.K., A.R. and R.C.; validation, K.K. writing—original draft preparation, K.K.; writing—review and editing, T.W.; supervision, J.W.; project administration, J.W.; funding acquisition, K.K., E.R. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Patient Room Description


```
{
  "patient_room": {
    "room_number": 1,
    "patient": {
      "first_name": "Jim",
      "last_name": "Smith",
      "pose": {
        "position": {
          "x": 3.5,
          "y": 6.8,
          "z": 1.0
      },
        "orientation": {
          "roll": 0,
          "pitch": 0,
          "yaw": 1.57
      }
    }
  },
  "nurse": {
    "first_name": "Kyungki",
    "last_name": "Kim"
  },
  "objects": [
    {
      "name": "water_cup",
      "pose": {
        "position": {
          "x": 6.77,
          "y": 4.17,
          "z": 1.0
        },
        "orientation": {
          "roll": 0,
          "pitch": 0,
          "yaw": 0
      }
    }
  },
    {
      "name": "bed",
      "pose": {
        "position": {
          "x": 2.5,
          "y": 6.5
        }
      },
      "boundary": [
        {"x": 2.5, "y": 6.5},
        {"x": 2.5, "y": 10.0},
        {"x": 4.0, "y": 10.0},
        {"x": 4.0, "y": 6.5}
      ]
    },
    {
      "name": "TV",
      "pose": {
        "position": {
          "x": 3.0,
          "y": 4.0,
          "z": 1.2
        }
      },
      "status": {
        "channel": 7,
        "is_on": true
      }
    }
  ],

"predetermined_locations": [
        {
          "description":
"patient_monitoring",
          "position": {
            "x": 1.5,
            "y": 7.0
          }
        },
        {
          "description":
"patient-robot_interaction",
          "position": {
            "x": 1.5,
            "y": 7.0
          }
        },
        {
          "description":
"robot_waiting_location",
          "position": {
            "x": 2.0,
            "y": 4.0
          }
        }
      ]
    }
  }
```

Appendix B. Task Specifications

Document Title: Robotic Health Attendant (RHA) Task Specifications

General Notes:
1. RHA situational understanding: RHA situational understanding describes how the RHA knows which tasks to perform in given situations. This is a trigger to select certain task specifications and perform actions as described in the task specifications.
2. RHA role: RHA role provides a general description for the expected roles.

Task Specification 1: Fetch Resources
- Description: Nurses or patients need items for patient care (e.g., water, medications, or medical supplies).

- RHA Role: The RHA should promptly respond to requests for fetching items, navigate to the required locations, and retrieve the necessary resources efficiently.

- Situational Understanding: Triggered by commands received from nurses or patients requesting specific items.

- Actions to Complete Operation:
Action 1. Navigate to item location: The RHA will navigate to the location where the requested item is located using the skill navigateToItemLocation, with the first input argument being the name of the object and second input argument being the coordinates of the item location. Since this is 2D navigation, provide X, Y, and orientation of the object. Z should be 0.
Action 2. Pick item: The RHA will fetch the required item based on the request using the skill pickObject. The first input argument is the name of the object and second input argument is the coordinates of the item location.
Action 3. Navigate to where the item is needed: The RHA will navigate to a designated location (e.g., patient monitoring, patient-robot interaction) where the item needs to be delivered using the skill navigateToLocation, with the input argument being the coordinates of the location needing the item. Don't navigate directly to a person.
Action 4. Handover the item: The RHA will hand over the item to the intended recipient using the skill releaseObject. No input argument is needed for this action.
Note: All these four actions must be carried out sequentially, with each step executed after the previous one. The robot has to submit a prompt saying the previous action is completed, before giving another action to the robot.

Appendix C. Interaction Guidelines

Introduction of roles
- I am a patient care robot operating in patient room, termed Robotic Health Attendant (RHA). Each task should be completed according to a corresponding task specification. I rely on your (ChatGPT's) instructions to understand the situation and follow the task specifications.

Notes for RHA-ChatGPT Interaction
1. A task is typically composed of one or more actions and each action is performed by executing robot skills. I (RHA) will submit prompts when I started and completed actions. You (ChatGPT) should provide responses. ChatGPT can provide two types of responses in json format following the structure shown in response examples below.
2. All the responses must be only in json format strictly following the examples. If you need to explain something, include that in the "description" in the json.
2. Type 1: When I complete an action, provide a json response for the required next action.
3. Type 2: While I am performing 1 or more actions, provide a json response indicating I am busy (status should be "busy"). An example can be found below.
4. In case of Type 2 responses or any other responses not triggering actions should call "no_action"
5. General communication with people starting with "Information:" or "Communication:" should not use json. Provide your answers in natural language format.
6. Entre response should in { } like the examples

```
# Response examples

good response example 1 (type 1):
{
  "actions": [
    {
      "description": "Navigate to the location of requested item",
      "function_called": "navigateToItemLocation",
      "arguments": [
        {"name": "item_name", "value": "water_cup"},
        {"name": "x", "value": 0.0},
        {"name": "y", "value": 0.0},
        {"name": "z", "value": 0.0},
        {"name": "roll", "value": 0},
        {"name": "pitch", "value": 0},
        {"name": "yaw", "value": 0}
      ]
    }
  ]
}

good response example 2 (type 1):
{
  "actions": [
    {
      "description": "Navigate to the designated location",
      "function_called": "navigateToLocation",
      "arguments": [
        {"name": "location_name", "value": "waiting area"},
        {"name": "x", "value": 0.0},
        {"name": "y", "value": 0.0},
        {"name": "z", "value": 0.0},
        {"name": "roll", "value": 0},
        {"name": "pitch", "value": 0},
        {"name": "yaw", "value": 0}
      ]
    }
  ]
}

good response example 3 (type 1):
{
  "actions": [
    {
      "description": "Pick up the requested item",
      "function_called": "pickObject",
      "arguments": [
        {"name": "object_name", "value": "medications"},
        {"name": "x", "value": 6.77},
        {"name": "y", "value": 4.17},
        {"name": "z", "value": 1.0},
        {"name": "roll", "value": 0},
        {"name": "pitch", "value": 0},
        {"name": "yaw", "value": 0}
      ]
    }
  ]
}
```

good response example 4 (type 1):

```
{
  "actions": [
    {
      "description": "Release the water cup",
      "function_called": "releaseObject",
      "arguments": [
        {"name": "object_name", "value": "water_cup"}
      ]
    }
  ]
}
```

good response example 5 (type 2):

```
{
  "status": "busy",
  "message": "No additional action required at the moment.",
  "actions_in_progress": [
    {
      "description": "The robot is currently navigating to the specified location.",
      "function_called": "noAction",
      "arguments": []
    }
  ]
}
```

good response example 6 (type 2):

```
{
  "status": "communicating",
  "message": "Communication such as patient status and information.",
  "actions_in_progress": [
    {
      "description": "The robot is currently describing...",
      "function_called": "noAction",
      "arguments": []
    }
  ]
}
```

bad response example 1 (type 1): This is wrong because part of the response is not in json.

Great! Now let's go to the next step, picking up the water cup.

```
{
  "actions": [
    {
      "description": "Pick up the requested item",
      "function_called": "pickObject",
      "arguments": [
        {"name": "object_name", "value": "water_cup"},
        {"name": "x", "value": 6.77},
        {"name": "y", "value": 4.17},
        {"name": "z", "value": 1.0},
        {"name": "roll", "value": 0},
        {"name": "pitch", "value": 0},
        {"name": "yaw", "value": 0}
      ]
    }
  ]
}
```

Appendix D. Robot Profile


```
{
  "Robot_ID": "rha_unmc_1",
  "Robot Type": "robotic health attendant",
  "Tasks": "follow task specification fetch Resources version 1",
  "Skill": [
     "navigateToLocation",
     "navigateToItemLocation",
     "readyToPick",
     "moveArmToObject",
     "pickObject",
     "releaseObject",
     "summarizePatientInformation",
     "transcribeAndRecord",
     "summarizeConversation",
     "waitForRequest",
     "noAction"
  ]
}
```


References

1. NIOSH Risk Factors for Stress and Burnout. Available online: https://www.cdc.gov/niosh/healthcare/risk-factors/stress-burnout.html (accessed on 3 June 2024).
2. NIOSH Risk Factors for Healthcare Workers. Available online: https://www.cdc.gov/niosh/healthcare/risk-factors/index.html (accessed on 3 June 2024).
3. The Harvard Gazette Doctors Not the Only Ones Feeling Burned Out. Available online: https://news.harvard.edu/gazette/story/2023/03/covid-burnout-hitting-all-levels-of-health-care-workforce/ (accessed on 4 June 2024).
4. American Hospital Association (AHA) Fact Sheet: Strengthening the Health Care Workforce. Available online: https://www.aha.org/fact-sheets/2021-05-26-fact-sheet-strengthening-health-care-workforce (accessed on 4 June 2024).
5. Manyika, J.; Chui, M.; Miremadi, M.; Bughin, J.; George, K.; Willmott, P.; Dewhurst, M. A Future That Works: AI, Automation, Employment, and Productivity. *McKinsey Glob. Inst. Res. Tech. Rep.* **2017**, *60*, 1–135.
6. Ranasinghe, R.; Dantanarayana, L.; Tran, A.; Lie, S.; Behrens, M.; Liu, L. Smart Hoist: An Assistive Robot to Aid Carers. In Proceedings of the 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV), Singapore, 10–12 December 2014; pp. 1285–1291.
7. Mukai, T.; Hirano, S.; Nakashima, H.; Kato, Y.; Sakaida, Y.; Guo, S.; Hosoe, S. Development of a Nursing-Care Assistant Robot RIBA That Can Lift a Human in Its Arms. In Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010—Conference Proceedings, Taipei, Taiwan, 18–20 October 2010.
8. Abubakar, S.; Das, S.K.; Robinson, C.; Saadatzi, M.N.; Cynthia Logsdon, M.; Mitchell, H.; Chlebowy, D.; Popa, D.O. ARNA, a Service Robot for Nursing Assistance: System Overview and User Acceptability. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Hong Kong, China, 20–21 August 2020.
9. Ding, J.; Lim, Y.J.; Solano, M.; Shadle, K.; Park, C.; Lin, C.; Hu, J. Giving Patients a Lift—The Robotic Nursing Assistant (Rona). In Proceedings of the IEEE Conference on Technologies for Practical Robot Applications, TePRA, Woburn, MA, USA, 14–15 April 2014.
10. Baltazar, A.R.; Petry, M.R.; Silva, M.F.; Moreira, A.P. Autonomous Wheelchair for Patient's Transportation on Healthcare Institutions. *SN Appl. Sci.* **2021**, *3*, 354. [CrossRef] [PubMed]
11. Shafii, N.; Farias, P.; Sousa, I.; Sobreira, H.; Reis, L.P.; Moreira, A.P. Autonomous Interactive Object Manipulation and Navigation Capabilities for an Intelligent Wheelchair. In Proceedings of the Progress in Artificial Intelligence: 18th EPIA Conference on Artificial Intelligence, EPIA 2017, Porto, Portugal, 5–8 September 2017; Proceedings 18; pp. 473–485.
12. Singapore-MIT Alliance for Research and Technology (SMART) Fact Sheet: Smart Self-Driving Wheelchair. Available online: https://portal.smart.mit.edu/images/pdf/news/2017/SMART_Factsheet_selfdriving_wheelchair_May2017.pdf (accessed on 25 October 2024).
13. Swisslog healthcare Relay Hospital Delivery Robot. Available online: https://www.swisslog-healthcare.com/en-gb/products/transport/relay (accessed on 19 October 2023).
14. Diligent Robotics Care Is a Team Effort. Available online: https://www.diligentrobots.com/moxi (accessed on 19 October 2023).
15. XENEX Light Strike Germ-Zapping Robots. Available online: https://xenex.com/light-strike/ (accessed on 19 October 2023).
16. surfacide Helios UVC Disinfection System. Available online: https://www.surfacide.com/helios (accessed on 19 October 2023).

17. Fischinger, D.; Einramhof, P.; Papoutsakis, K.; Wohlkinger, W.; Mayer, P.; Panek, P.; Hofmann, S.; Koertner, T.; Weiss, A.; Argyros, A.; et al. Hobbit, a Care Robot Supporting Independent Living at Home: First Prototype and Lessons Learned. *Rob. Auton. Syst.* **2016**, *75*, 60–78. [CrossRef]
18. Miseikis, J.; Caroni, P.; Duchamp, P.; Gasser, A.; Marko, R.; Miseikiene, N.; Zwilling, F.; De Castelbajac, C.; Eicher, L.; Fruh, M.; et al. Lio-A Personal Robot Assistant for Human-Robot Interaction and Care Applications. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5339–5346. [CrossRef] [PubMed]
19. Holland, J.; Kingston, L.; McCarthy, C.; Armstrong, E.; O'dwyer, P.; Merz, F.; McConnell, M. Service Robots in the Healthcare Sector. *Robotics* **2021**, *10*, 47. [CrossRef]
20. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report. *arXiv* **2023**, arXiv:2303.08774.
21. Zeng, A.; Attarian, M.; Ichter, B.; Choromanski, K.; Wong, A.; Welker, S.; Tombari, F.; Purohit, A.; Ryoo, M.; Sindhwani, V.; et al. Socratic Models: Composing Zero-Shot Multimodal Reasoning with Language. *arXiv* **2022**, arXiv:2204.00598.
22. Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Fu, C.; Gopalakrishnan, K.; Hausman, K.; et al. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In Proceedings of the Machine Learning Research, Seattle, WA, USA, 30 November–1 December 2023; Volume 205.
23. Huang, W.; Xia, F.; Xiao, T.; Chan, H.; Liang, J.; Florence, P.; Zeng, A.; Tompson, J.; Mordatch, I.; Chebotar, Y.; et al. Inner Monologue: Embodied Reasoning through Planning with Language Models. In Proceedings of the Machine Learning Research, Seattle, WA, USA, 30 November–1 December 2023; Volume 205.
24. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An Open-Source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
25. Bed Transport Assistance Robot—Time Medical. Available online: https://www.time-medical.com/bed-transport-assistance-robot (accessed on 28 March 2024).
26. Bloss, R. Mobile Hospital Robots Cure Numerous Logistic Needs. *Ind. Robot. Int. J.* **2011**, *38*, 567–571. [CrossRef]
27. Evans, J.M. HelpMate(R): An Autonomous Mobile Robot Courier for Hospitals. In Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, Munich, Germany, 12–16 September 1994; Volume 3.
28. Bacik, J.; Durovsky, F.; Biros, M.; Kyslan, K.; Perdukova, D.; Padmanaban, S. Pathfinder-Development of Automated Guided Vehicle for Hospital Logistics. *IEEE Access* **2017**, *5*, 26892–26900. [CrossRef]
29. Carreira, F.; Canas, T.; Silva, A.; Cardeira, C. I-Merc: A Mobile Robot to Deliver Meals inside Health Services. In Proceedings of the 2006 IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand, 7–9 June 2006.
30. Simon, M. This Incredible Hospital Robot Is Saving Lives. Also, I Hate It. Available online: https://www.wired.com/2015/02/incredible-hospital-robot-saving-lives-also-hate/ (accessed on 3 June 2024).
31. Panasonic Group Panasonic Autonomous Delivery Robots—HOSPI—Aid Hospital Operations at Changi General Hospital. Available online: https://news.panasonic.com/global/topics/4923 (accessed on 3 June 2024).
32. Chanprakon, P.; Sae-Oung, T.; Treebupachatsakul, T.; Hannanta-Anan, P.; Piyawattanametha, W. An Ultra-Violet Sterilization Robot for Disinfection. In Proceedings of the 5th International Conference on Engineering, Applied Sciences and Technology, ICEAST 2019, Luang Prabang, Laos, 2–5 July 2019.
33. Demaitre, E. Third Generation of UV-C Disinfection Robot Available from UVD Robots. Available online: https://www.therobotreport.com/third-generation-uv-c-disinfection-robot-available-uvd-robots/ (accessed on 9 June 2024).
34. Kwon, J. New and Improved Germ Zapping Robots Used in Local Hospitals. Available online: https://spectrumnews1.com/ca/southern-california/coronavirus/2021/03/31/new-and-improved-germ-zapping-robots-used-in-local-hospitals (accessed on 9 June 2024).
35. Koh, D. Omron Launches UVC Disinfection Robot Targeting COVID-19 Virus. Available online: https://www.mobihealthnews.com/news/asia/omron-launches-uvc-disinfection-robot-targeting-covid-19-virus (accessed on 9 June 2024).
36. Jasmin and Lavender Disinfection Robots Launched by Geek+ Worldwide in Response to Pandemic. Available online: https://www.therobotreport.com/jasmin-lavender-geekplus-launches-two-disinfection-robots-worldwide/ (accessed on 9 June 2024).
37. Pandey, A.K.; Gelin, R. A Mass-Produced Sociable Humanoid Robot: Pepper: The First Machine of Its Kind. *IEEE Robot. Autom. Mag.* **2018**, *25*, 40–48. [CrossRef]
38. PAL Robotics TIAGo Base Delivery in Hospitals. Available online: https://pal-robotics.com/blog/tiago-delivery-impact-hospitals-covid19/ (accessed on 30 July 2024).
39. Tamantini, C.; Scotto Di Luzio, F.; Cordella, F.; Pascarella, G.; Agro, F.E.; Zollo, L. A Robotic Health-Care Assistant for COVID-19 Emergency: A Proposed Solution for Logistics and Disinfection in a Hospital Environment. *IEEE Robot. Autom. Mag.* **2021**, *28*, 71–81. [CrossRef]
40. PAL Robotics OpenDR Project: Enhancing Healthcare Robotics in Trials with the TIAGo Robot. Available online: https://pal-robotics.com/blog/opendr-project-enhancing-healthcare-robotics-with-the-tiago-robot/ (accessed on 30 July 2024).
41. Ding, Y.; Zhang, X.; Amiri, S.; Cao, N.; Yang, H.; Kaminski, A.; Esselink, C.; Zhang, S. Integrating Action Knowledge and LLMs for Task Planning and Situation Handling in Open Worlds. *Auton. Robots* **2023**, *47*, 981–997. [CrossRef]
42. Tenorth, M.; Beetz, M. KnowRob: A Knowledge Processing Infrastructure for Cognition-Enabled Robots. *Int. J. Rob. Res.* **2013**, *32*, 566–590. [CrossRef]

43. Rovida, F.; Crosby, M.; Holz, D.; Polydoros, A.S.; Großmann, B.; Petrick, R.P.A.; Krüger, V. SkiROS. In *Robot Operating System (ROS)*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 121–160.
44. Perera, V.; Soetens, R.; Kollar, T.; Samadi, M.; Sun, Y.; Nardi, D.; van de Molengraft, R.; Veloso, M. Learning Task Knowledge from Dialog and Web Access. *Robotics* **2015**, *4*, 223. [CrossRef]
45. Amiri, S.; Bajracharya, S.; Goktolgal, C.; Thomason, J.; Zhang, S. Augmenting Knowledge through Statistical, Goal-Oriented Human-Robot Dialog. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Macau, China, 3–8 November 2019.
46. Jiang, Y.; Walker, N.; Hart, J.; Stone, P. Open-World Reasoning for Service Robots. In Proceedings of the International Conference on Automated Planning and Scheduling, ICAPS, Berkeley, CA, USA, 11–15 July 2019.
47. Singh, I.; Blukis, V.; Mousavian, A.; Goyal, A.; Xu, D.; Tremblay, J.; Fox, D.; Thomason, J.; Garg, A. ProgPrompt: Generating Situated Robot Task Plans Using Large Language Models. In Proceedings of the IEEE International Conference on Robotics and Automation, London, UK, 29 May–2 June 2023.
48. Vemprala, S.; Bonatti, R.; Bucker, A.; Kapoor, A. Chatgpt for Robotics: Design Principles and Model Abilities. *arXiv* **2023**, arXiv:2306.17582. [CrossRef]
49. Liang, J.; Huang, W.; Xia, F.; Xu, P.; Hausman, K.; Ichter, B.; Florence, P.; Zeng, A. Code as Policies: Language Model Programs for Embodied Control. In Proceedings of the IEEE International Conference on Robotics and Automation, London, UK, 29 May–2 June 2023.
50. Gazebo Robot Simulation Made Easy. Available online: http://gazebosim.org/ (accessed on 28 November 2021).
51. ROS.org Amcl. Available online: http://wiki.ros.org/amcl (accessed on 28 November 2021).
52. Chitta, S. Moveit!: An Introduction. In *Robot Operating System (ROS)*; Studies in Computational Intelligence; Springer: Cham, Switzerland, 2016; Volume 625. [CrossRef]
53. Hao Yang GitHub—Pgm_map_creator: Create Pgm Map from Gazebo World File for ROS Localization. Available online: https://github.com/hyfan1116/pgm_map_creator (accessed on 28 November 2021).

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.