International Journal of Social Robotics (2026) 18:15
https://doi.org/10.1007/s12369-025-01335-1

KG-Retailbot: A Knowledge Graph-Based Chatbot for Explaining Robotic Scenario Information in a Retail Setting

Ke Xu[1] · Sen Yuan[2] · Sanja Dogramadzi[1] · Carlos Hernández Corbato[3]

Received: 29 June 2024 / Revised: 11 September 2025 / Accepted: 19 December 2025

Abstract

Robots are now pervasive, leveraging their automation capabilities to assist humans across a diverse range of tasks. Nevertheless, end-users may have a limited understanding of the robot's operation and typically assume a passive role when interacting with the robot performing a particular task. In this study, we address the critical need for effective explainability in human-robot interaction. By comparing different methods of explaining robotic scenario information to end-users, the proposed methodologies use a labelled property graph-based chatbot that adheres to the IEEE Robotics Ontology Standards. In this study, we designed two virtual robotic scenarios and simulated their information flow using the Robot Operating System. A between-subjects experiment was conducted where participants engaged with the system through various interaction methods to understand the two scenarios. These methods included real-time Linux Command Line Interface outputs, querying a chatbot, exploring knowledge graphs, or a combination of chatbot and knowledge graphs. The study findings suggest that both the knowledge graphs and the chatbot significantly enhance the system's explainability compared to a simple Linux terminal information output. Moreover, utilizing knowledge graphs alongside the chatbot has received better subjective evaluations concerning metrics such as clarity, usability, and robustness. This research made contributions towards the development of standardised labelled property graphs for representing scenario information in language-based human-robot interaction. The experiment design and evaluations also provided a solution for assessing the explainability of task-oriented dialogue systems both subjectively and objectively.

Keywords Ontology · Knowledge representation · Knowledge graph · Chatbot · Human-robot interaction · Rasa

1 Introduction

With the rapid advancement of robotics and intelligent agents, the demand for automated behaviours has increased dramatically. However, as agents take on decision-making roles in automated systems, their explainability becomes critical for enabling end-users to make informed and accountable actions [1] in human-robot interaction (HRI) contexts. Explainable systems empower users to interact effectively by understanding the underlying mechanisms and decisions behind automated behaviours.

Despite progress in algorithmic transparency, current solutions often struggle to fully address end-user needs for practical, scenario-specific explanations in robotic systems. This gap becomes particularly significant when users are required to trust and collaborate with robotic systems in complex, dynamic environments. In such scenarios, transparent explanations play a key role by bridging the gap between system complexity and user understanding. They

✉ Sen Yuan
s.yuan-3@tudelft.nl

Ke Xu
kxu30@sheffield.ac.uk

Sanja Dogramadzi
s.dogramadzi@sheffield.ac.uk

Carlos Hernández Corbato
C.H.Corbato@tudelft.nl

[1] School of Electrical and Electronic Engineering, The University of Sheffield, Sheffield, UK

[2] Department of Microelectronics, Delft University of Technology, Delft, The Netherlands

[3] Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands

Published online: 30 January 2026

enable users to better comprehend, predict, and validate the actions of robotic agents, making explainability a critical component for improving both user trust and system adoption. To address this gap, dialogue systems (DSs), leveraging natural language, serve as a powerful tool. It enables interactive communication and clarification between users and robots, providing a natural and intuitive way to deliver scenario-specific explanations. This approach not only supports real-time information exchange but also serves as a bridge between complex system functionalities and user understanding, ensuring that robotic systems remain both accessible and comprehensible to diverse users.

To address the challenges of delivering explainable and scenario-specific information, this paper introduces a knowledge graph (KG)-based framework for dialogue-driven interaction. By integrating structured knowledge representation (KR) with natural language processing (NLP), the proposed system is designed to enhance user comprehension of both static and runtime robotic scenarios, with a case study in a retail setting.

Our contributions can be summarised as follows.

1. **Efficient Knowledge Representation:** We propose the Integrated Ontology for Robotics and Automation (IORA)-labelled property graph (LPG) schema, derived from several standardized IEEE ontologies, to effectively represent robotic scenario knowledge for task-oriented applications.
2. **KG-Integrated Dialogue System:** We develop a task-oriented chatbot that integrates KG-based structured knowledge representation with natural language interaction, enabling real-time scenario-specific explanations in a retail environment.
3. **Comprehensive System Evaluation:** We conduct a between-subject study to assess the system's effectiveness, comparing different interaction methods through both subjective user feedback and objective performance metrics, ensuring a well-rounded evaluation of usability and information retrieval efficiency.

2 Related Work

Explainability in Robotic Systems Explainability has become a critical focus in robotic systems, particularly in enabling users to trust and collaborate with autonomous agents [2]. Existing research has primarily focused on algorithmic transparency, with efforts to elucidate model structures and decision processes to mitigate the opacity of black-box algorithms [3, 4]. While effective in mitigating the opacity of black-box algorithms, these approaches often lack direct relevance to end-user interaction, especially in scenario-specific contexts.

The concept of "explanation-for-trust" [5] highlights the importance of revealing a system's internal mechanisms to enhance user trust and understanding. In the context of robotic systems, this is particularly critical, as users often interact with complex, autonomous agents whose decisions and behaviours directly impact task outcomes. Providing transparent explanations ensures that users can comprehend these mechanisms, fostering a deeper understanding and confidence in the system's operations. However, achieving this level of transparency requires practical and interactive tools tailored to the unique needs of specific robotic environments.

Dialogue Systems in HRI DSs, for instance, can serve as an effective tool for achieving explainability in HRI, as verbal interaction is widely recognized as the most natural and effective mode of communication [6]. These systems take advantage of advancements in NLP to facilitate seamless interaction between users and robots. Broadly, DSs in HRI can be categorized into two main applications: grounding natural language commands into robotic actions and serving as conversational assistants for chat or support purposes.

A key application of DSs is interpreting user instructions and mapping them to specific robotic actions. Early research explored task-specific NLP techniques such as deep semantic role labelling [7] and conditional random fields [8], to extract structured task-related information from user inputs [9, 10]. These methods allowed robots to execute commands accurately in constrained environments, laying the foundation for more advanced systems. Recent advances have incorporated large language models (LLMs) to enhance the flexibility of command interpretation. For example, Koubaa et al. [11] leveraged LLMs with prompt engineering to generate executable robotic tasks from unstructured user inputs. However, this system relies on predefined ontologies to align commands with robotic actions [12], limiting the adaptability in unanticipated scenarios. Therefore, the need for more flexible and standardized KR methods should be considered to support various task-oriented applications.

Beyond task execution, DSs function as conversational assistants, enabling robots to engage in meaningful interactions for social and support purposes. These systems focus on conversational flow, making them ideal for applications like companionship and user assistance. For instance, Grassi et al. [13] used Google Dialogflow and its Natural Language API to capture user intents and facilitate conversational interactions. Similarly, Fujii et al. [14, 15] developed a Rasa [16]-based dialogue system that transformed the Nao Robot into an interactive dining companion. PAL Robotics utilized the ROS4HRI standard [17], a framework for developing interactive robots, to integrate the Rasa

framework with social robots for elderly care [18]. This implementation demonstrated how dialogue systems, when built on standardized frameworks, can enhance accessibility and engagement in non-task-specific contexts. However, these systems are primarily designed for general conversational assistance and lack the capacity to provide detailed, scenario-specific explanations, which are crucial in complex robotic environments.

Knowledge Representation in HRI The effective representation of scenario information is essential for enabling DSs to deliver meaningful and explainable interactions in HRI contexts. Task ontologies have been employed to organize domain knowledge into structured hierarchies. Jokinen et al. [19] employed a task ontology to structure caregiving tasks, providing users with detailed instructions for eight common caregiving actions. Similarly, the CARESSES framework [20] extended conversational diversity by employing runtime-extensible ontologies to facilitate culturally adaptive dialogues across diverse user backgrounds [13]. However, these approaches primarily support task planning or conversational flow management, rather than serving as dedicated knowledge bases (KBs) capable of storing scenario information. Consequently, they fall short in equipping DSs with the comprehensive understanding needed to explain and interpret robotic scenarios effectively.

Building on the foundational use of ontologies, recent research has shifted towards KGs for representing domain knowledge in a more dynamic and scalable manner. Ait-Mlouk et al. demonstrated the use of linked data [21]-based chatbot to convert natural language queries into SPARQL commands to retrieve information from KBs such as DBpedia and Wikidata [22]. Meanwhile, Wilcock integrated Neo4j KGs [23] with Rasa-based dialogue systems, showcasing improved dialogue flexibility in applications like tourism [24, 25] and later deploying the system on the Furhat Robots [26] for practical evaluation [27]. Although these studies demonstrate the utility of KGs in general-purpose applications, they do not fully meet the demands of task-oriented HRI, particularly in addressing challenges such as runtime task execution and runtime adaptability. Advancing this area requires the development of standardized and flexible KGs that can enable DSs to deliver scenario-specific explanations while adapting to diverse and evolving robotic environments.

3 Fundamentals

In this paper, a KG-based chatbot presents a promising method for implementing the system's explainability of real-time robotics scenarios. Generally, there are two approaches to building KGs: top-down and bottom-up [28]. The bottom-up strategy relies on automated extraction technologies to derive concepts and relationships from semi-structured data, prioritizing those with higher confidence levels for inclusion in the KB. This approach demands consistent access to high-quality data sources to manage and update the schema effectively. In contrast, the top-down approach entails defining the ontology and data schema of the KG as a prerequisite for incorporating entities to the KB. This necessitates a group of experts possessing a profound comprehension of the domain-specific knowledge hierarchy.

Utilizing standardised ontologies for the KG-based chatbot is imperative to ensure semantic coherence, enabling efficient understanding and response generation across diverse user interactions. Recent research proposed several standardised ontologies to model terminologies in the robotics domain: Core Ontology for Robotics and Automation (CORA) [29]-related ontologies (containing Suggested Upper Merged Ontology (SUMO) [30]-CORA, CORAX, PRARTS and POS) [31], ERAS ontology [32] and Task ontology (TO) [33].

- SUMO-CORA: SUMO-CORA is a comprehensive top-level ontology designed to define the fundamental ontological categories in the real world.
- CORA & CORAX & POS & RPARTS: CORA includes three main concepts: *RobotGroup*, *Robot* and *RobotSystem*, while CORAX defines some not-so-generic but essential robotic concepts. POS defines concepts related to objects' pose, position, and orientation properties. RPARTS comprises concepts representing specific devices that can constitute robot parts.
- ERAS: ERAS ontology considers concepts regarding the ethical usage of robotic techniques based on CORA ontology.
- TO: Task ontology focuses on the task implementation terminology as an extension of CORA ontology.

Due to these well-defined ontologies, the top-down approach was appropriate for constructing knowledge hierarchies in this work. Once the schema of KG is defined by ontologies, data can be stored graphically. Resource Description Framework (RDF) [34]/Web Ontology Language (OWL) [35] (e.g., Jena [36]) and LPG [37, 38] databases (e.g., Neo4j) are emerging technologies for storing graph-structured data [39]. The LPG format offers a more compact representation of multiple properties using arrays compared to RDF [40]. Additionally, direct relationships between two entities are established, aligning closely with human KR patterns of real-world information. Therefore, we opted for LPG as the KG storage approach for our system, leveraging its efficient and intuitive structures.

4 Method

4.1 System Architecture

To achieve the goal of delivering scenario-specific explanations in HRI, we develop KG-Retailbot, a knowledge-driven dialogue system using a structured multi-module framework. Our approach integrates task-oriented KGs with a natural language DS to facilitate intuitive and accurate information retrieval. As depicted in Fig. 1, the proposed system comprises three core modules: the Robotic System (RS) module, responsible for collecting real-time robotic execution data; the KG module, which organizes and stores structured task-related knowledge; and the DS module, enabling knowledge retrieval for user interaction. These modules operate in a structured workflow to bridge the gap between raw robotic execution data and human-interpretable scenario explanations.

The RS module acquires execution data from a runtime Robot operating system (ROS) environment, including task goals, planned actions, and execution status, retrieved via the ROS Action Servers and ROS Parameter Servers. This builds upon the work of Corrado et al. [41], which employed active inference for dynamic task execution. To integrate these processes with structured KGs, we developed a dedicated ROS node, *create_dynamic_kg*, which subscribes to the relevant servers and continuously updates scenario information.

The extracted robotic scenario data is structured and stored in the KG module using a task-oriented LPG schema within a Neo4j database management system (DBMS). This transformation follows a structured pipeline (detailed in Sect. 4.2), enabling explicit representation of items, tasks, and execution concepts in robotics environments. The Py2neo library [42] facilitates dynamic updates as new data arrives. Users can also explore the structured KGs via the Neo4j Browser, providing an interpretable view of the stored scenario knowledge.

The DS module, built using Rasa, leverages a trained NLU model to process user queries. It performs intent classification and entity recognition before invoking action servers to retrieve relevant task-related knowledge from the KG module. These servers interface with the Neo4j DBMS to extract structured information, which is then processed

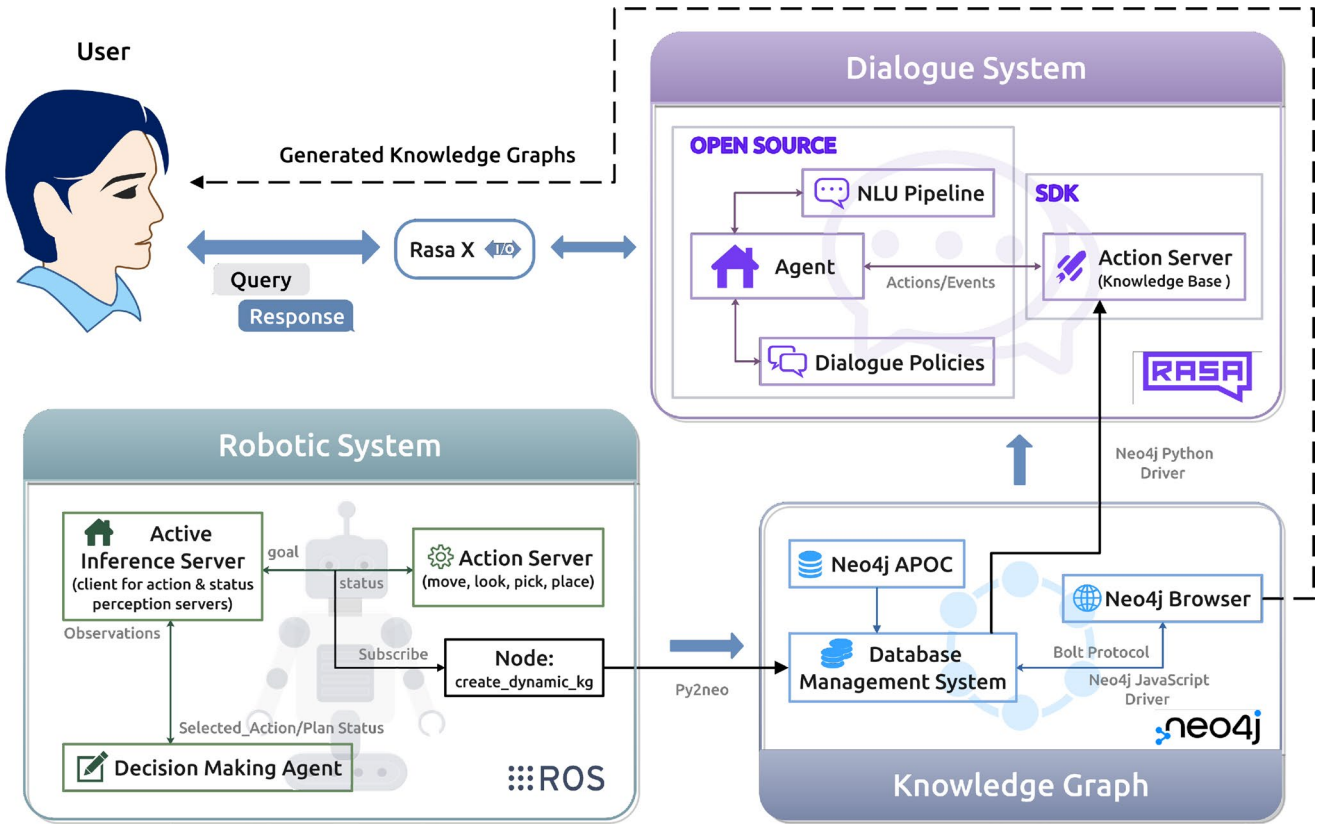


Fig. 1 System architecture of KG-Retailbot: two interactive methods are illustrated by black lines: chatbot (solid line) and KGs (dashed line). The directions of the arrows at the end of the black lines indicate the primary data flow between three modules: 1) robotic data in RS module is extracted as structural knowledge in Neo4j DBMS using the Py2neo library. 2) the generated KGs are accessed by Rasa action servers through Neo4j python Driver. 3) Rasa X provides a communication interface for users. 4) users can view generated KGs directly through Neo4j browser

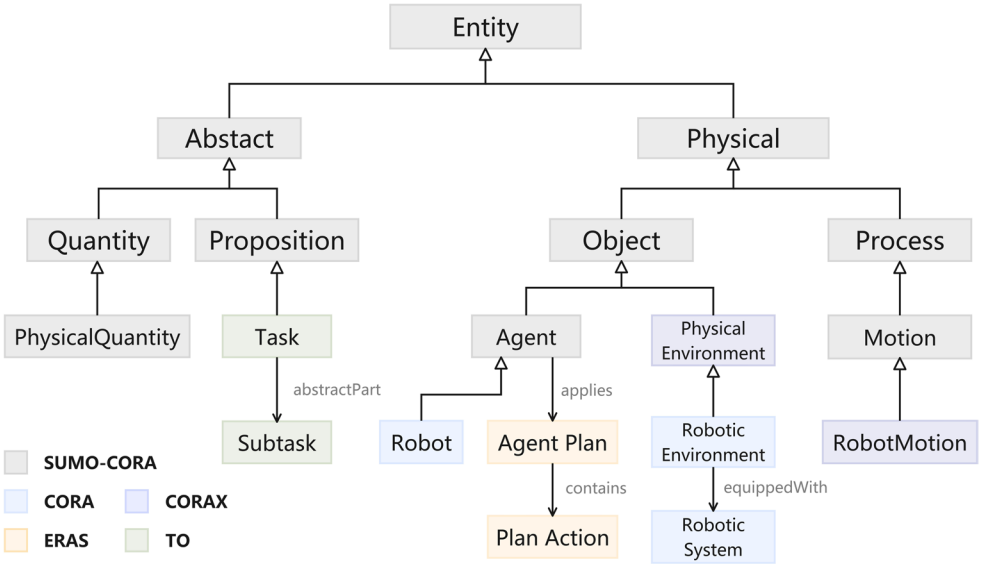


Fig. 2 Overview of main concepts in Integrated ontology for robotics and automation (IORA)

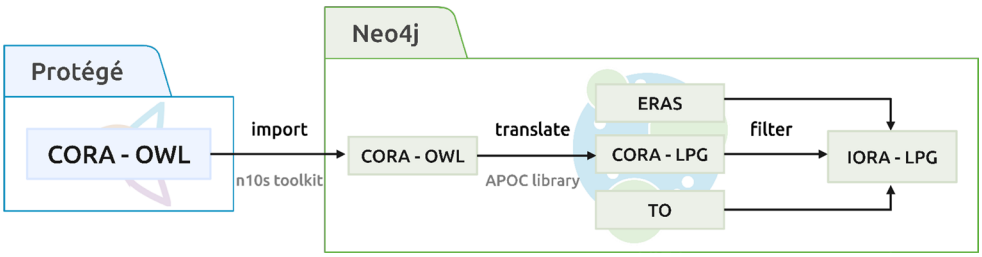


Fig. 3 Pipeline to adapt OWL-based ontologies to LPG-based schema

into scenario-specific natural language explanations. Dialogue Policies control the dialogue flow, while the Agent manages system integration and I/O operations, ensuring smooth user-robot communication.

By integrating structured KR with natural language interaction, this framework provides an explainable interface for users to query and interpret robotic operations. The KG, DS pipelines are further detailed in the following section.

4.2 Knowledge Graph

To structure knowledge for task-oriented robotic scenarios, we develop a domain-specific Integrated Ontology for Robotics and Automation (IORA). The ontology integrates concepts from CORA-related, ERAS, and Task ontologies mentioned in Sect. 3, as outlined in Fig. 2.

Ontology-based representations provide a formal structure for capturing domain knowledge. However, traditional OWL-based ontologies are not well suited for intuitive querying and adaptation of runtime scenarios. To enhance usability, we transform the OWL ontologies into a LPG schema, enabling efficient knowledge retrieval and task inference. This transformation is implemented using Neo4j[1], a property graph database optimized for structured queries and semantic reasoning.

As shown in Fig. 3, we first import CORA-related ontologies [43][2] into Neo4j DBMS using the neosemantics (n10s) toolkit [44], producing an initial graph structure with 147 nodes and 249 relationships. To enable efficient querying and reasoning, we transform OWL classes and properties into LPG nodes and relationships using the Neo4j APOC library. This process restructures the ontology into a task-oriented schema while preserving its semantic relationships, as detailed in the following paragraph. To refine the representation, we manually filter concepts relevant to HRI scenarios, integrating key elements from ERAS and Task ontologies. The final IORA-LPG consists of 34 nodes and 64 relationships, forming a structured schema for task-oriented robot scenarios.

To transform OWL ontologies into a task-oriented LPG schema, we implement a structured four-step process, as outlined in Algorithm 1, ensuring consistency and usability in downstream applications.

Step 1: Standardizing Imported Tags OWL ontologies imported via the n10s library in RDF/XML format generate predefined labels, relationships, and properties that require renaming to align with the OWL syntax. Node labels (e.g.

[1] Neo4j: Version 4.4.12 was used in this study.

[2] CORA-related ontologies can be accessed by its open-source http://GitHubrepository.

Algorithm 1 Transformation Steps

Input: OWL ontologies
Output: LPG schema
Step *Rename auto-generated tags during OWL files import to conform with OWL syntax definitions.*
Node labels:
(a) $(n4sch\_\_Class) \rightarrow$ class description $\langle owl : Class \rangle$
(b) $(n4sch\_\_Relationship) \rightarrow$ object property $\langle owl : ObjectProperty \rangle$
Relationship types:
(a) $[n4sch\_\_SCO] \rightarrow$ subclass axioms $\langle rdfs : subClassOf \rangle$
(b) $[n4sch\_\_SPO] \rightarrow$ property axioms $\langle rdfs : subPropertyOf \rangle$
(c) $[n4sch\_\_SCO\_RESTRICTION] \rightarrow$ property restrictions $\langle owl : Restriction \rangle$
Property keys:
(a) $\langle owl : Class/ObjectProperty \rangle.n4sch\_\_name \rightarrow \langle owl : Class/ObjectProperty \rangle.name$
(b) $\langle owl : Class/ObjectProperty \rangle.n4sch\_\_propCharacteristics$
$\rightarrow \langle owl : Class/ObjectProperty \rangle.propCharacteristics$
Step *Transform property restrictions* $\langle owl : Restriction \rangle$ *between classes* $\langle owl : Class \rangle$ *into properties of nodes* $\langle owl : Class/ObjectProperty \rangle$.
(a) *Value constraints*: $\langle owl : Restriction \rangle.restrictionType \rightarrow \langle owl : ObjectProperty \rangle.restrictionType$
(b) *Cardinality constraints*: $\langle owl : Restriction \rangle.cardinalityVal \rightarrow \langle owl : ObjectProperty \rangle.cardinalityVal$
Step *Project classes* $\langle owl : ObjectProperty \rangle$ *and relationships* $\langle rdfs : subPropertyOf \rangle$ *onto a subgraph to preserve subPropertyOf axioms before transforming* $\langle owl : ObjectProperty \rangle$ *into edges in LPG.*
Step *Transform main RDFS construct-relationships* $\langle owl : ObjectProperty \rangle$ *between classes* $\langle owl : Class \rangle$ *into LPG constructs.*
$(\langle owl : Class \rangle) \leftarrow [\langle rdfs : range \rangle]$ - $(\langle owl : ObjectProperty \rangle)$ - $[\langle rdfs : domain \rangle] \rightarrow (\langle owl : Class \rangle)$ **to** $(\langle owl : Class \rangle)$ - $(\langle owl : ObjectProperty \rangle) \rightarrow (\langle owl : Class \rangle)$

class description) and relationship types (e.g. property axioms) are reformatted to ensure semantic clarity for further transformation.

Step 2: Converting Property Restrictions OWL property restrictions, which define constraints on object properties, are incorporated into the LPG schema as node attributes. Value constraints and cardinality constraints are extracted and stored directly within the corresponding object attributes, maintaining the intended logical structure.

Step 3: Preserving Hierarchical Relations To retain the hierarchical organization of properties, $\langle rdfs : subPropertyOf \rangle$ axioms are projected onto a subgraph before the $\langle owl : ObjectProperty \rangle$ are transformed into edges. This preserves the inheritance structure between properties, allowing for more structured reasoning.

Step 4: Generating Graph Relationships Finally, $\langle owl : ObjectProperty \rangle$ and their associated domain-range relationships are converted into LPG constructs. Instead of treating object properties as independent nodes, they are transformed into direct edges between $\langle owl : Class \rangle$ nodes, facilitating efficient traversal and knowledge retrieval.

4.3 Dialogue system

In AIRLab Delft[3], a mobile-based robotic manipulator is used to perform pick-and-place product tasks in a retail setting, as shown in Fig. 4. Using the previously described LPG-based framework, we developed a dialogue system tailored to this specific robotic scenario.

The system is implemented using the Rasa framework[4], enabling users to query both static product properties (e.g., mass and position) and runtime robot task information, such as task details and execution status. Rasa was chosen for its modular architecture, supporting customizable NLU pipelines, deep learning-based dialogue policies, and flexible action services for adaptive responses.

To ensure robust NLU within the constraints of limited hand-crafted training data, we employed multiple components. SpacyNLP [45] was integrated for tokenization and word embedding via SpacyTokenizer and SpacyFeaturizer. The DIETClassifier [46] was used for both intent classification and entity extraction, using transformer-based embeddings to improve generalization in limited training samples.

[3] This research was partially supported by Ahold Delhaize. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

[4] Rasa: Rasa 3.1.4 and Rasa X 1.1.3 were used in this study.

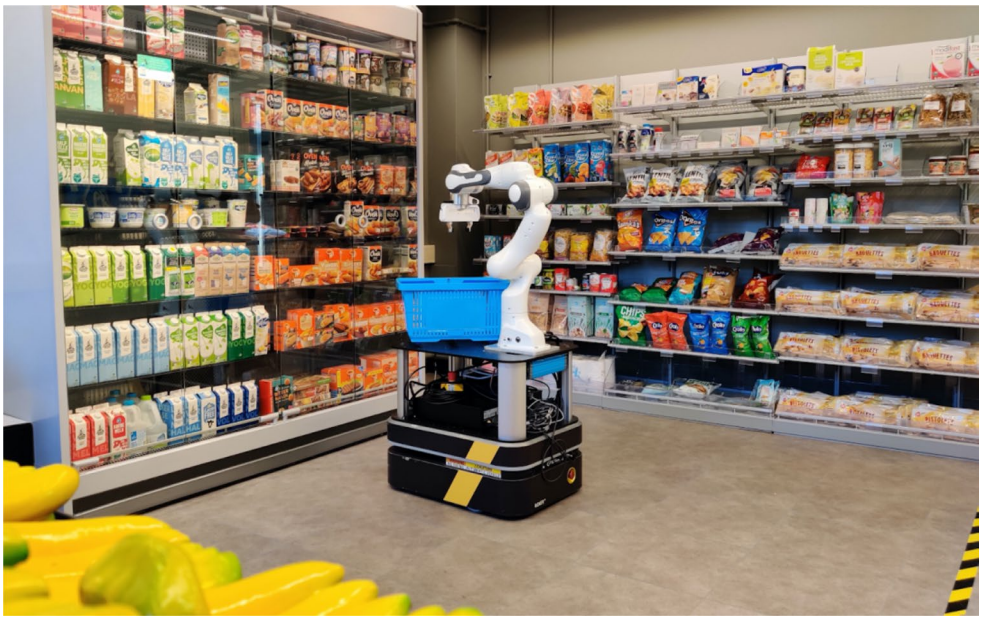


Fig. 4 The retail environment in AIRLab Delft

Table 1 NLU training data and response examples: square brackets [ ] indicate the text that represents an entity, while parentheses ( ) specify the entity type being labeled. The bracketed ellipsis [...] is used to indicate omitted information while preserving the sentence structure

Intent	NLU Training example	Response Example
query_environment	Give me some background on AIRLab.	The AI for Retail Lab is a joint industry lab.
introduce_chatbot	Who am I talking with?	My name is Iris, a chatbot assistant for you […]
chatbot_capability	What can you do?	You can ask me the following things […]
query_product_in_env	What [items](object_type) are in the environment?	There are […] products in the environment: […]
query_product_property	Tell me the [mass](attribute) of [yoghurt](object).	Sure, [object]'s [attribute] is […]
query_specific_product	Does the environment contain [flower](object)?	It seems the environment doesn't contain […]
	Tell me [more](property) about [yogurt](object).	No problem, let me tell you some properties […]
	What type of [properties](property) can I ask?	You could ask properties of products within […]
query_product_location	Where could I find [juice](object)?	I think [object] is located at […] on the [furniture].
query_product_furniture	What products are on [shelf_1](furniture)?	On [furniture], you can find product: […]
query_specific_task	I want to know about [task2](task).	[task]: the robot moves to the place where […]
query_current_task	What is the current task?	Sure! The Robot is performing [task]: […]
query_previous_task	Give me an overview of previous tasks.	[task]: moved to […]. [task]: picked [object].
query_current_action	What are you doing now?	No problem! The robot is placing [object].

Additionally, the FallbackClassifier handled low-confidence utterances, ensuring system robustness by triggering clarification requests or fallback responses. To enhance entity recognition consistency, the EntitySynonymMapper was used to normalize synonymous terms, mapping variations of user input to unified entity labels.

Dialogue management relies on a combination of policies to maintain coherent and context-aware interactions. The MemoizationPolicy enabled the system to recall frequently occurring dialogue paths, allowing predefined conversations to be handled efficiently. TEDPolicy, a Transformer-based model [47], generalized beyond memorized paths, predicting system responses based on contextual information. The RulePolicy defined fixed behaviours, such as fallback handling, ensuring that ambiguous or unsupported queries triggered appropriate system responses.

When the user submits a query, the dialogue system processes it through NLU pipeline and dialogue policies to determine the appropriate response. If knowledge retrieval is needed, Rasa action servers trigger Cypher queries [48] via the Py2neo library to extract relevant information from the Neo4j DBMS. The retrieved data is then formatted into natural language explanations and presented to the user.

The predefined user intents and their corresponding response examples are summarized in Table 1. Intents related to static environmental information are listed above the horizontal line, while those concerning runtime robot task information are listed below. Furthermore, seven entity types (*object_type*, *object*, *attribute*, *property*, *furniture*, *task*, and *subtask*) were defined to ensure the system accurately extracts and stores task-relevant entity values. The

full set of functional action servers used to retrieve back-end updated KGs is detailed in Table 9 in Appendix A.

5 Experiments

A between-subject experiment is designed to evaluate the effectiveness and performance of our proposed KG-Retailbot in assisting users to comprehend two distinct retail scenarios: Scenario 1 (static) and Scenario 2 (runtime). In Scenario 1, a set of predefined retail products, along with their attributes such as location and quality, were stored in the product KG. In Scenario 2, we did not simulate the physical motion of the robot within the ROS environment. Instead, our focus was on simulating the information flow transmission of the RS module, as detailed in Sect. 4.1. Utilizing ROS topics, services, and actions, runtime task goals and actions planned by the decision-making agent, as well as the robot's motion status, were continuously updated and stored in the task KG. This approach enabled us to replicate the runtime interaction and data flow of the robotic system without requiring actual robot movements, effectively creating a "black box" retail scenario for the users.

In the two simulated scenarios, participants could only access and comprehend scenario information through different interaction methods provided at the front-end within a limited time frame, including Linux Command Line Interface (CLI), KGs and the chatbot. The effectiveness of these interaction methods was assessed by the accuracy of participants' responses in information-recall questionnaires. Higher accuracy in these questionnaires indicates that our system can more efficiently convey scenario information to users within the designed retail environment. Additionally, a separate questionnaire was employed to capture users' subjective evaluations of the chatbot's performance. Some assumptions were held and given here when designing the experiment:

- **Assumption 1**: Participants with similar backgrounds were recruited for this study. They were provided with a comprehensive description of the experiment and the associated questionnaires prior to participation. Given this standardized preparation, we assume that the participants' levels of expertise in ROS or robotics do not introduce any bias in the results.
- **Assumption 2**: Participants were required to recall retail product and robot task information obtained during their interactions within a limited timeframe. All participants were given the same amount of time to interact and to complete the information recall questionnaires. We assume that individual differences in memory will not significantly affect the results due to the relatively short duration of the interaction and recall period.

5.1 Scenario Design

5.1.1 Static Scenario (Scenario 1)

In the first scenario, participants need to acquire static information about the products stored in a predefined KG. This KG contains four key properties (*id*, *mass*, *grasping_position*, *position*) for six common products: *hagelslag* (chocolate milk in Dutch supermarket), *yoghurt*, *milk*, *tea_box* and *ice_cream*. Here, *grasping_position* refers to the robot's target coordinates for initiating a grasp action, while position refers to the product's actual location in the scenario. Both properties are represented as Cartesian coordinates (x, y, z) and stored in the KG. Product-related concepts (*Object*, *PositionMeasure*, *PositionRegion*, *PoseMeasure*) and their relationships (*pose*, *positionAt*, *inPR*) from IORA-LPG were used to model this knowledge. For instance, as illustrated in Fig. 5, a fragment of product information displayed in the Neo4j browser shows detailed properties of a *tea_box*, including its mass (0.5 kg) and its position in the *basket*. During the experiment, participants could directly interact with the product KG to obtain information through the Neo4j browser. Besides, they could query the chatbot using natural language sentences, such as *'Tell me the mass of milk.'* to obtain these properties.

5.1.2 Runtime Scenario (Scenario 2)

The second scenario involves three robotic tasks, each with a different final state: 1) the robot moves to the location of the *milk*, 2) the robot picks up the *hagelslag* box, and 3) the robot places the *tea_box* in the basket. Task-related concepts from the IORA-LPG (*Agent*, $AgentPlan$, $PlanAction$, *Task*, *Subtask*), along with their relationships (*applies*, *contains*, $is\_implemented\_by$, *abstractPart*) are used to store the details of the plan and the execution status of these tasks. For instance, the pink and purple boxes in Fig. 6 illustrate task_3, involving three subtasks (actions): *move to reach tea_box*, *pick tea_box* and *place it in basket*, ultimately achieving the final state where the *tea_box* is in the *basket*. During the experiment, participants could access dynamically-generated KGs via the Neo4j browser to obtain detailed task information, including task goals, product states, and the planned action sequences. Additionally, they could review the CLI outputs shown in Fig. 7, or query the chatbot using natural language phrases like *'What are you doing now?'*. These methods allowed participants to efficiently comprehend the robotic tasks and their progress.

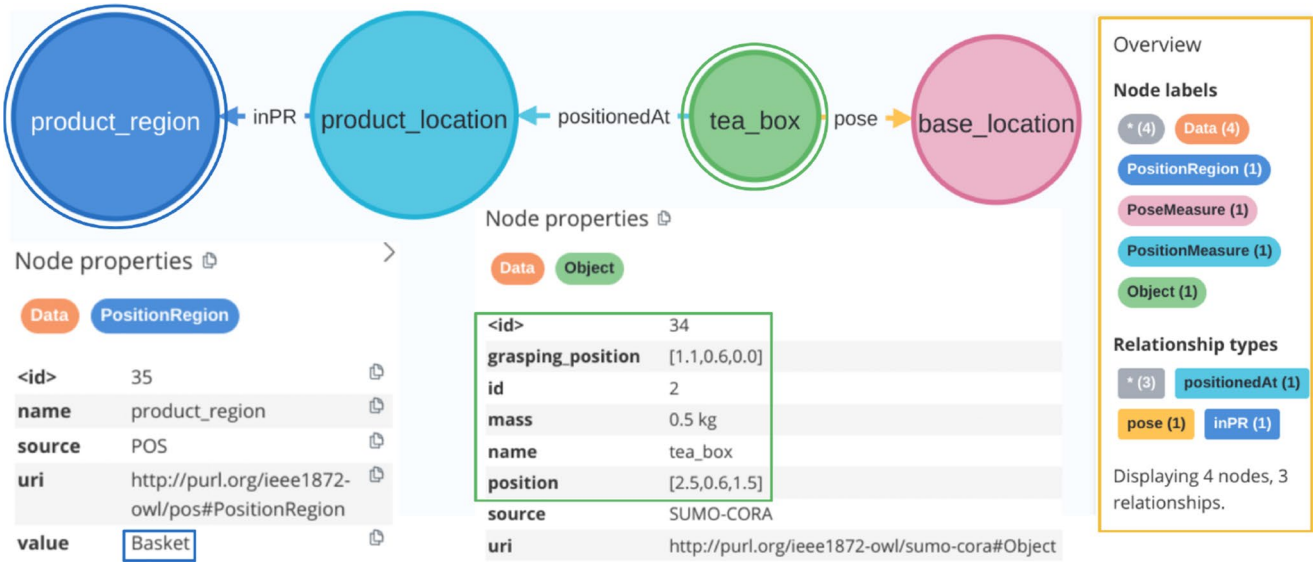


Fig. 5 One fragment of the product KG in Scenario 1: In Neo4j browser, a "node" represents an entity with attributes, and an "edge" signifies a directional relationship between nodes with properties. The "overview" in the right bar provides a comprehensive visualization of the graph's structure. For example, node labels and relationship types of Scenario 1 are summarized in the right yellow box. In the center, the green node contains all properties of the *tea_box*, while other nodes store its location information

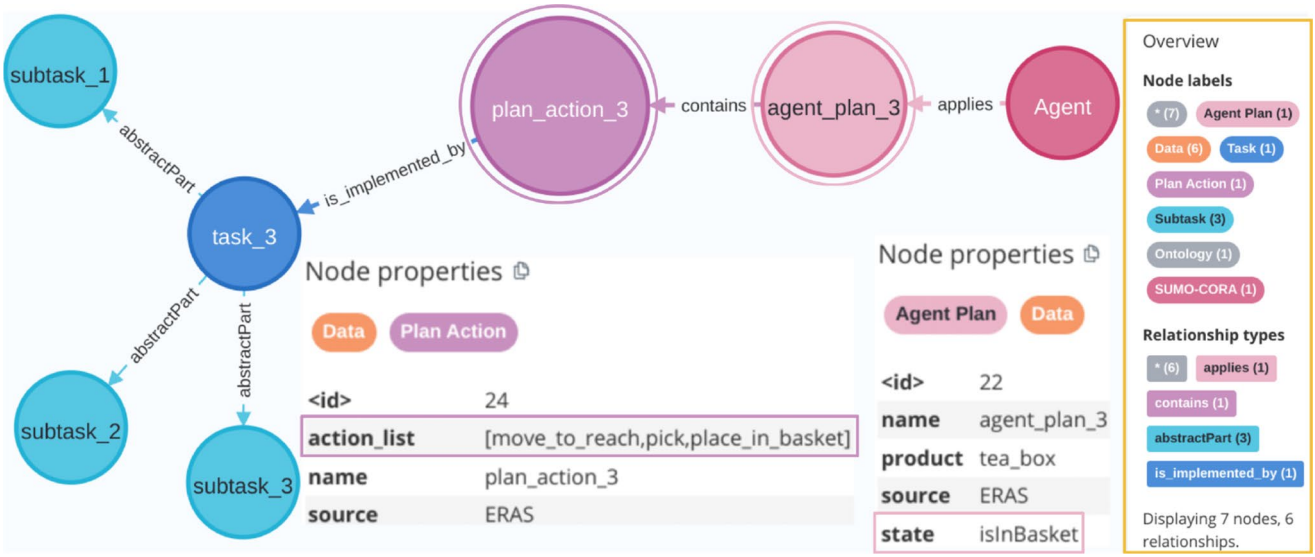


Fig. 6 One fragment of the task KG in Scenario 2: node labels and relationship types are detailed in the yellow box on the right. In the center, pink and purple nodes store plan details made by the decision-making agent, while blue nodes track the execution status of robot tasks and actions

5.2 Participants

A between-subject study was conducted to mitigate potential carryover effects from learning and fatigue [49]. Four groups of participants, each consisting of 5–10 students and researchers with relevant backgrounds in Robotics and Electronic Engineering, were recruited from the university. These groups were tasked with interacting with the system to comprehend the information presented in the two scenarios described previously. Each group was assigned a different method of interaction, as illustrated by the experimental scenarios involving the chatbot and KGs shown in Fig. 8. Participants were subsequently provided with specific questionnaires designed to evaluate both their understanding of the scenarios and the performance of the system. The questionnaires included the product information-recall assessment (Questionnaire 1), task information-recall assessment (Questionnaire 2), and the chatbot satisfaction assessment (Questionnaire 3), as detailed in the Tables 2, 3 and 4, respectively.

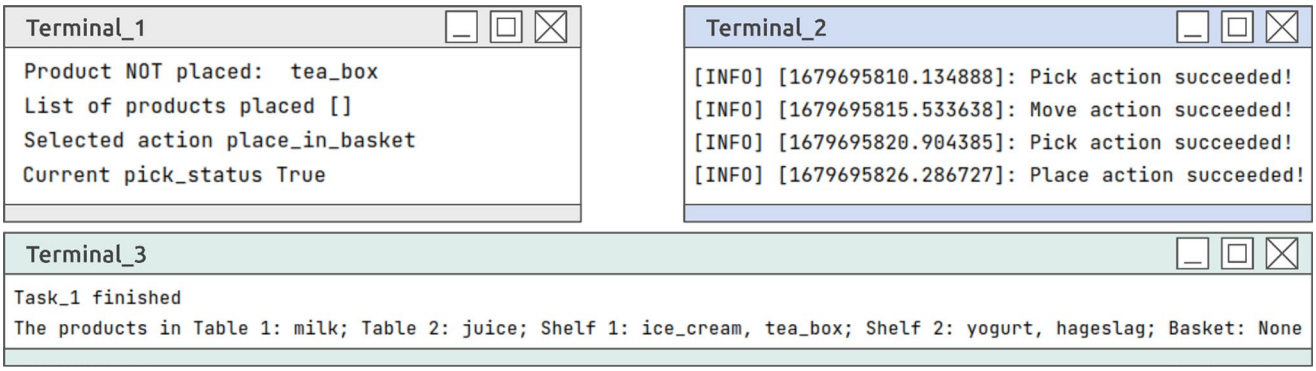
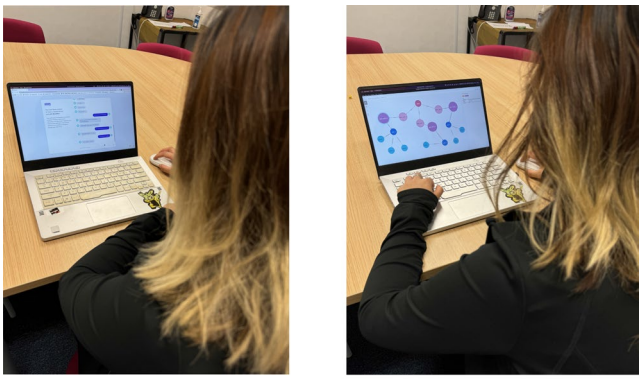


Fig. 7 One fragment of Linux CLI outputs in Scenario 3: Terminal_1 shows the action planed by the decision-making agent; Terminal_2 informs participants of completed actions; Terminal_3 prints completed task and updates product locations



(a) Scenario with Chatbot (b) Scenario with KG

Fig. 8 Experimental scene showing interaction with (**a**) the chatbot &; (**b**) knowledge graphs

Table 2 Questionnaire 1 - product information recall of Scenario 1

Item	ID	mass	grasping_position	position
Milk				
Tea box				
Hageslag				
Yogurt				
Juice				
Ice cream				

- **Group 1**: This group focused on viewing the real-time Linux CLI outputs to understand the robotic tasks in Scenario 2. They were allocated 5 minutes for this task and completed Questionnaire 2, which assessed their recall of the task-related information.
- **Group 2**: Participants interacted exclusively with the chatbot for a total of 10 minutes, equally divided between Scenario 1 and Scenario 2. They used the chatbot to gather product and task information and completed all three questionnaires, evaluating their understanding of the scenarios and the chatbot's performance.
- **Group 3**: This group only accessed the static and runtime KGs to comprehend Scenario 1 and Scenario 2, with a total interaction time of 10 minutes. Afterwards, they completed Questionnaire 1 and Questionnaire 2 to evaluate their understanding based on the KG data.
- **Group 4**: Participants in this group used both the chatbot and the static and runtime KGs. They interacted with the chatbot in the same manner as Group 2 and additionally explored the KGs for 5 minutes. They completed the same set of questionnaires as Group 2 to assess their understanding and experience with the system.

6 Results

A total of 25 participants were recruited and randomly assigned to the different groups as detailed in Table 5. The distribution was: seven participants were in the Linux CLI group, four in the Chatbot group, seven in the KG group, and seven in the Chatbot+KG group. While participants were randomly assigned, their stated preferences were also considered in order to improve task engagement. Figure 9 presents examples of chat history from Group 2 and Group 4 during their interactions in Scenario 1 and Scenario 2. These examples illustrate how participants used the chatbot to query about product and task information in the experiment.

To comprehensively analyze the experimental data, we focused on three main aspects:

- Consistency Assessment: We employed the Intraclass Correlation Coefficient (ICC) [50] to evaluate the consistency of information recall accuracy within each group. ICC is a statistical measure that assesses how strongly units in the same group resemble each other. This approach allowed us to assess how consistently participants in each group could accurately recall information, indicating the reliability of the interaction methods used.
- System Effectiveness Evaluation: We assessed the effectiveness of the system by comparing how accurately participants recalled detailed product and task information

Table 3 Questionnaire 2 - task information recall of Scenario 2

1	What is robot's first task?	◯ Move		◯ Pick		◯ Place		◯ Not sure
2	What is robot's second task?	◯ Move		◯ Pick		◯ Place		◯ Not sure
3	What is robot's third task?	◯ Move		◯ Pick		◯ Place		◯ Not sure
4	Which products are robot's first task?	◯ Milk	◯ Tea box	◯Hageslag	◯ Yogurt	◯ Juice	◯ Ice cream	◯ Not sure
5	Which products are robot's second task?	◯ Milk	◯ Tea box	◯Hageslag	◯ Yogurt	◯ Juice	◯ Ice cream	◯ Not sure
6	Which products are robot's third task?	◯ Milk	◯ Tea box	◯Hageslag	◯ Yogurt	◯ Juice	◯ Ice cream	◯ Not sure
7	Which products are on the Table 1 at the end?	◯ Milk	◯ Tea box	◯Hageslag	◯ Yogurt	◯ Juice	◯ Ice cream	◯ Not sure
8	Which products are on the Table 2 at the end?	◯ Milk	◯ Tea box	◯Hageslag	◯ Yogurt	◯ Juice	◯ Ice cream	◯ Not sure
9	Which products are on the Shelf 1 at the end?	◯ Milk	◯ Tea box	◯Hageslag	◯ Yogurt	◯ Juice	◯ Ice cream	◯ Not sure
10	Which products are on the Shelf 2 at the end?	◯ Milk	◯ Tea box	◯Hageslag	◯ Yogurt	◯ Juice	◯ Ice cream	◯ Not sure
11	What's your age	Please write your answer:						
level		1	2	3	4	5	6	7
12	What your professional level with ROS	◯	◯	◯	◯	◯	◯	◯
13	What your professional level with Knowledge graph	◯	◯	◯	◯	◯	◯	◯

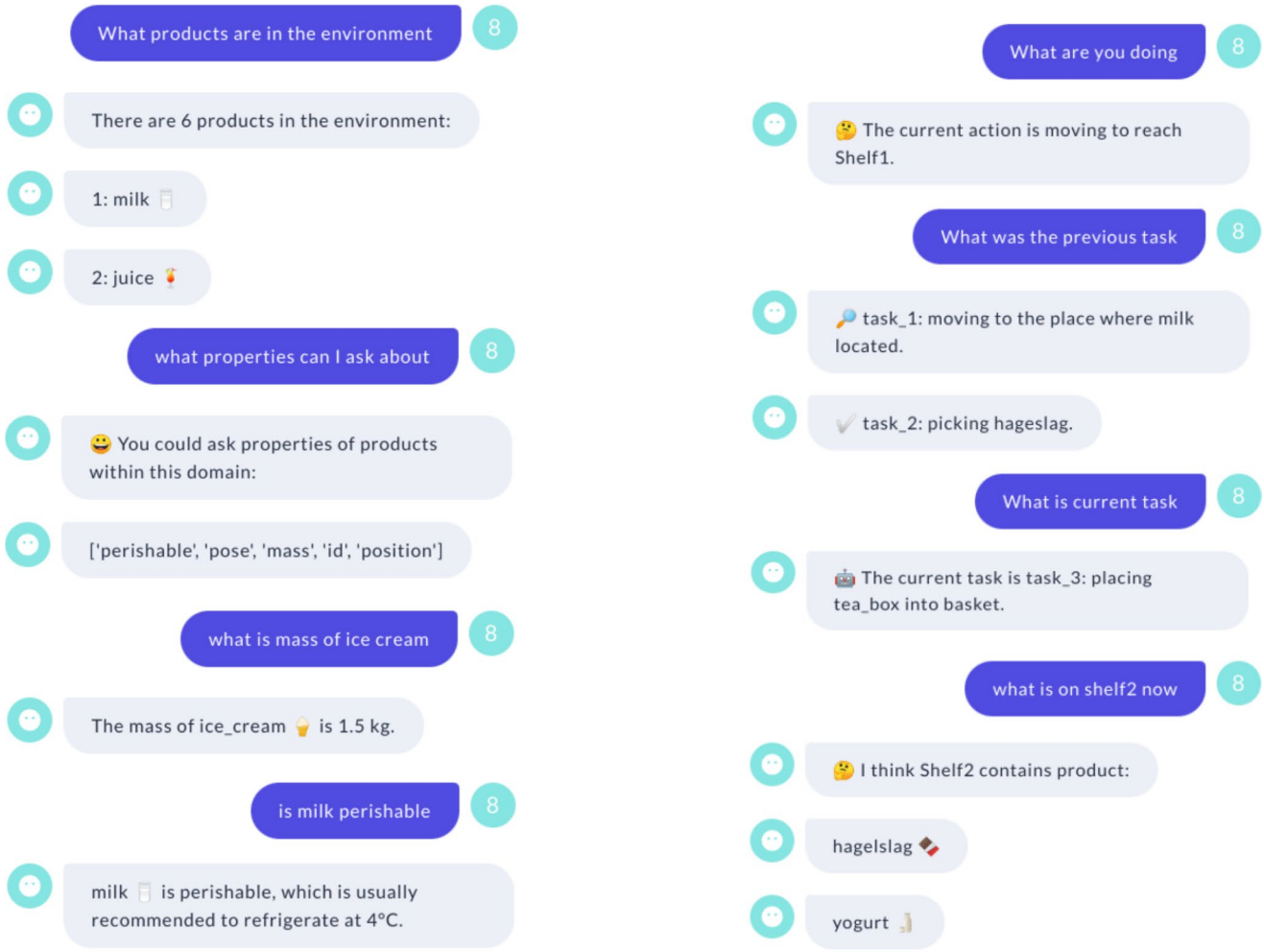
Table 4 Questionnaire 3 - chatbot evaluation

Question List		Strongly Disagree			Strongly Agree			
		1	2	3	4	5	5	7
Usability								
1	The chatbot responds too slowly.	◯	◯	◯	◯	◯	◯	◯
2	It was easy to lose track of where you are in the interaction.	◯	◯	◯	◯	◯	◯	◯
3	It is easy to learn how to use the chatbot.	◯	◯	◯	◯	◯	◯	◯
Clarity								
4	The chabot's responses were accurate.	◯	◯	◯	◯	◯	◯	◯
5	The chatbot didn't always do what I wanted.	◯	◯	◯	◯	◯	◯	◯
6	The chatbot was organized and logical.	◯	◯	◯	◯	◯	◯	◯
Naturalness								
7	The chatbot was understandable.	◯	◯	◯	◯	◯	◯	◯
8	The interaction with the chatbot was consistent.	◯	◯	◯	◯	◯	◯	◯
9	The chatbot used everyday words.	◯	◯	◯	◯	◯	◯	◯
Friendliness								
10	The chatbot's response sounded enthusiastic.	◯	◯	◯	◯	◯	◯	◯
11	I felt comfortable using the system.	◯	◯	◯	◯	◯	◯	◯
12	The chatbot seemed friendly.	◯	◯	◯	◯	◯	◯	◯
Robustness to misunderstandings								
13	I was able to recover easily from errors.	◯	◯	◯	◯	◯	◯	◯
14	The chatbot made a few errors.	◯	◯	◯	◯	◯	◯	◯
15	I felt in control of the interaction with the chatbot.	◯	◯	◯	◯	◯	◯	◯
Willingness to use the system again								
16	I would be likely to use this system again.	◯	◯	◯	◯	◯	◯	◯
17	The system was useful.	◯	◯	◯	◯	◯	◯	◯
18	The chatbot would help me be more productive.	◯	◯	◯	◯	◯	◯	◯

Table 5 Intraclass correlation coefficient analysis of information recall consistency across four interaction groups

Group(Num)	ICC	Confidence probability
CLI (7)	0.8998	$p < 4.61e-7$
Chatbot (4)	0.6917	$p<0.0034$
KG (7)	0.7242	$p < 3.07e-4$
KG+Chatbot (7)	0.8166	$p < 1.090e-6$

across different interaction methods for Scenario 1 and Scenario 2. This evaluation provided insights into which method or combination of methods most effectively supported users in understanding and retaining scenario-specific information.



(a) Chat example run of Scenario 1 (b) Chat example run of Scenario 2

Fig. 9 Chat examples to query (**a**) static information &; (**b**) runtime information.

- Chatbot Performance Evaluation: For the groups that interacted with the chatbot (Group 2 and Group 4), we used the PARAdigm for Dialogue System Evaluation PARADISE [51] framework to evaluate its performance. This framework uses subjective feedback from participants and objective measures like task success and dialogue efficiency to comprehensively evaluate the chatbot's effectiveness and user experience.

6.1 Consistency Assessment

To compute the ICC, we first calculated the accuracy of participants' responses for each attribute in the information-recall questionnaires. For example, the accuracy of the *ID* attribute in Table 2 was based on the number of correctly identified items. The *position* attribute was counted as correct only when all three discrete coordinates (x, y, z) matched the ground-truth values, as the task was to assess whether the system enabled users to obtain the exact positional information rather than approximate spatial proximity. For multi-answer questions in Table 3, accuracy was computed based on the number of correctly selected items (i.e., the intersection between the correct answers and the participant's responses). The resulting accuracy percentages for each attribute were then discretised into categorical scores to facilitate a more robust reliability analysis.

We used ICC(A, k), following the two-way random effects model with absolute agreement [52], to assess the consistency of group-level accuracy across conditions. This formulation was chosen because our objective was to determine whether different interaction groups, treated as independent raters, produced comparable accuracy scores for each question. Unlike models assessing rank-order consistency, ICC(A,k) explicitly captures agreement in actual values, making it suitable for evaluating agreement in recall

performance. The ICC was then computed based on the following formulation:

$$ICC(A,k) = \frac{MS_R - MS_E}{MS_R + \frac{MS_C - MS_E}{n} + \frac{(k-1)MS_E}{k}} \quad (1)$$

where MS_R is the mean square for subjects, MS_C indicates the mean square for answers, MS_E is the mean square for residual error, n is the number of subjects, and k is the number of answers per subject.

The detailed ICC results are presented in Table 5. The ICC values for the CLI and KG+Chatbot groups were both above 0.75 ($p<0.001$), indicating a high level of consistency in their ability to accurately recall the information presented in the scenarios. The KG group had a slightly low ICC of 0.72 ($p<0.001$), which still demonstrates good reliability in their recall performance. In contrast, the Chatbot group exhibited an ICC of 0.69 ($p<0.01$), suggesting moderate consistency in recall accuracy. The relatively lower ICC in the Chatbot group may be attributed to its smaller sample size, which can limit the precision and stability of the statistical estimates. For the KG group, the slightly reduced ICC could be due to the varied individual strategies participants employed in allocating their time between the static and runtime KGs within the 10-minute period. This variability in focus distribution might have introduced differences in their recall performance.

Overall, these ICC results demonstrate that the data collected within each group is consistent. This consistency is crucial as it ensures that the subsequent analysis of system effectiveness, based on the accuracy of information recall questionnaires, is built on robust and dependable data. Thus, we can confidently proceed with comparing the effectiveness of different interaction methods, knowing that the foundation of our data is statistically sound.

6.2 System Effectiveness

To evaluate the effectiveness of different interaction methods in assisting participants' comprehension of scenario knowledge, we conducted a statistical analysis on accuracy scores obtained from Questionnaire 1 (Scenario 1) and Questionnaire 2 (Scenario 2).

Since the questionnaires were intentionally designed to include both simple fact-retrieval and more complex multi-step reasoning questions, using equal weights would risk overvaluing trivial items and undervaluing more informative ones. Accuracy was computed by comparing participants' responses against the ground truth using a weighted sum approach to account for varying question difficulty levels. Rather than assigning question weights arbitrarily or uniformly, we adopted a data-driven weighting scheme informed by participants' response accuracy. This method ensures that questions with higher agreement (i.e., clearer and less ambiguous) contribute more to the final score, which reflects their informativeness in the experimental context. The weights are calculated as:

$$weight_i = \frac{1}{error\_rate_i + \epsilon} \quad (2)$$

Where $error\_ratio_i$ is the proportion of participants who answered questions i incorrect, ϵ is a smoothing constant and is set to 0.05.

The weights of questions in Scenario 1 are calculated as [0.3952, 0.2865, 0.1378, 0.1804], while the weights of the Scenario 2 are [0.1141, 0.1538, 0.1787, 0.1489 0.1340, 0.0744, 0.1960]. The final accuracy scores are presented in Fig. 10. Overall, each group performed better in Scenario 2 (runtime) compared to Scenario 1 (static). We attribute this difference to the inherent difficulty of the questionnaires rather than the effectiveness of the interaction methods, so this aspect will not be further discussed here.

A one-way ANOVA [53] was performed to compare accuracy scores across the four experimental groups. The results indicated a significant main effect of interaction method in both Scenario 1 ($F=4.25$, $p=0.035$) and Scenario 2 ($F=5.03$, $p=0.009$), suggesting that different interaction modalities influenced participants' ability to recall scenario information.

To further analyze group-wise differences, Tukey's HSD post-hoc test [54] was conducted. The results showed that in Scenario 1, the Chatbot+KG group ($\overline{X} = 0.6508$, $VAR=0.0996$) significantly outperform the KG group ($\overline{X} = 0.3100$, $VAR=0.2816$) ($p=0.029$). These results indicate that our KG-Retailbot significantly enhances users' understanding of static product information within a retail environment. However, the difference between the KG and Chatbot groups was not statistically significant ($p=0.63$), suggesting that the efficiency of inquiring about product properties using simple sentences is comparable to directly accessing the product KG. This result indicates that while the KG method has advantages in presenting complex, multi-layered runtime tasks, both methods are similarly effective in aiding participants to retrieve static information.

In Scenario 2, the Chatbot+KG group ($\overline{X} = 0.7217$, $VAR=0.1504$) outperform the CLI group ($\overline{X} = 0.3875$, $VAR=0.1048$) ($p=0.011$) as expected. No significant differences were observed between the other groups with ($E(p)=0.50$). This was probably due to the small amount of experimental data collected plus some product properties that were not queried by the participants but were still included in the calculations for a fair comparison, making the differences smaller. Therefore, the effect

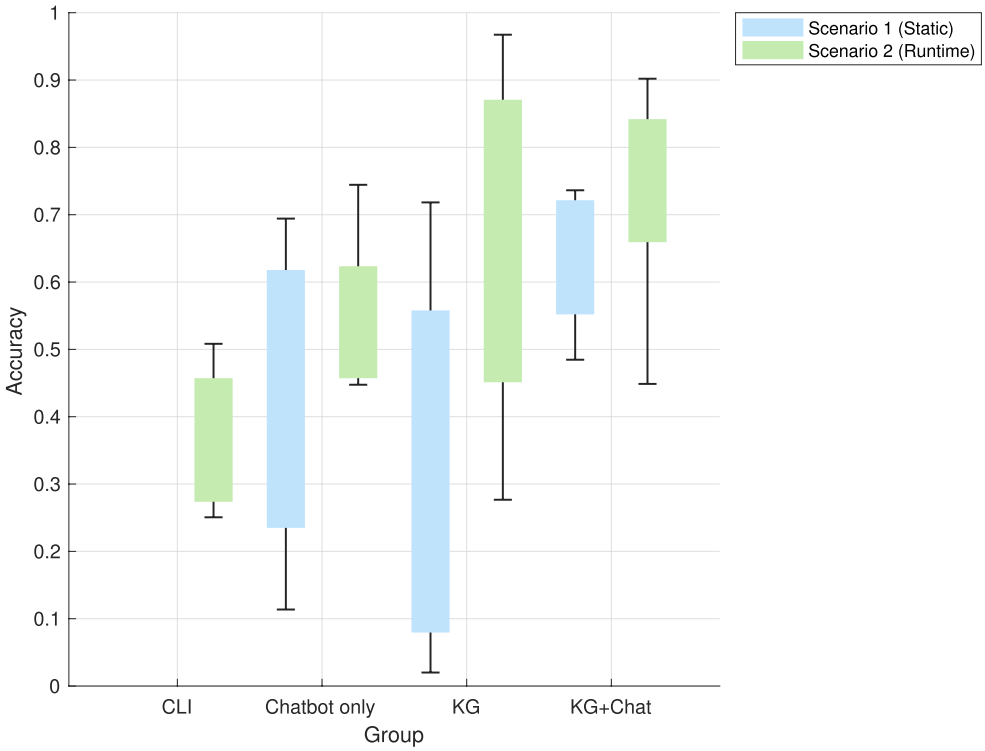


Fig. 10 Comparison of information recall accuracy across four interaction groups

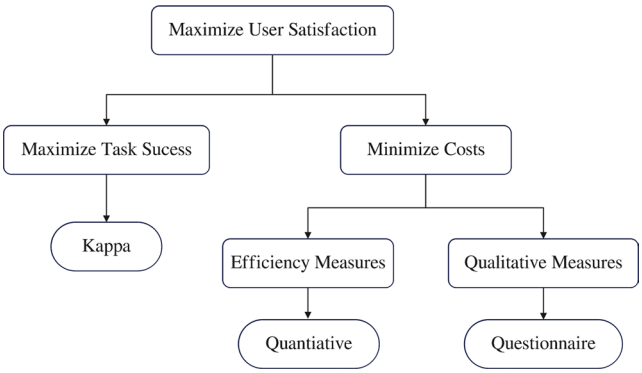


Fig. 11 Structure of PARADISE Framework [51]

size η^2 [55] is calculated considering the small amount of samples, and it equals to 0.3606 and 0.4181 in Scenario 1 and 2 respectively, demonstrating the large effect between the groups.

6.3 Chatbot Performance

The PARADISE framework was adopted to evaluate the chatbot performance by quantifying user satisfaction, which is contributed by two types of factors: task success and dialogue costs, shown in Fig. 11. Task success was measured by evaluating how effectively the chatbot understood and processed user input. Higher accuracy rates indicate a more effective chatbot in understanding and processing user inputs, leading to successful task completion. Dialogue costs were evaluated through two types of measures: efficiency and qualitative aspects. Efficiency costs refer to the resource consumption required to complete a task, such as the number of dialogue turns, and other quantifiable metrics. Lower efficiency costs indicate that the chatbot can achieve task completion with minimal resource expenditure, reflecting higher efficiency. Qualitative costs refer to the quality of conversational content, including aspects such as clarity and user engagement. Qualitative costs were assessed using questionnaires in which participants rated their conversational experience with the chatbot. Lower qualitative costs correspond to higher user satisfaction with the interaction. By combining these metrics, the PARADISE framework provides a comprehensive evaluation of the chatbot's performance, capturing both the success in task completion and the overall user experience during the interaction.

6.3.1 Task Success

Task success was quantified using the accuracy rates of intent recognition and entity recognition. These metrics measure how accurately the chatbot identifies the user's intent and correctly recognizes relevant entities within the conversation. It can be calculated by the Kappa coefficient κ [56], as shown in Eq. (3).

$$k = \frac{P(A) - P(E)}{1 - P(E)} \tag{3}$$

Table 6 Categories used in confusion matrices for intent and entity recognition across two scenarios

Attribute	Scenarios 1 (Static)	Scenarios 2 (Runtime)
Intent	query_product_property; query_perishable; query_product; query_specific_product; query_environment; chatbot_capability; introduce_chatbot; nlu_fallback	query_current_task; query_previous_task; query_specific_task; query_furniture; query_current_action; query_previous_action; query_predict; query_product_location; nlu_fallback
Entity	object; attribute; object_type; None; property	task; furniture; object; None; object_type

Table 7 Evaluation of task success using Kappa

Scenario	Type	Mean	Variance
Scenario 1 (Static)	Intention	0.7295	0.0280
	Entity	0.9058	0.0285
Scenario 2 (Runtime)	Intention	0.8208	0.0133
	Entity	0.8912	0.0068

Table 8 Metrics for dialogue costs in chatbot evaluation

Metric	Type	Data Collection Method
Total number of user/system turns	Efficiency	Quantitative
Total elapsed time per turn	Efficiency	Quantitative
Number of re-prompts	Qualitative	Quantitative
Number of inappropriate responses	Qualitative	Quantitative
Usability	Qualitative	Questionnaire
Clarity	Qualitative	Questionnaire
Naturalness	Qualitative	Questionnaire
Friendliness	Qualitative	Questionnaire
Robustness to misunderstandings	Qualitative	Questionnaire
Willingness to use the system again	Qualitative	Questionnaire

Here, $P(A)$ represents the observed accuracy, which is the proportion of times the chatbot correctly identifies intents and entities in all attempts. $P(E)$ represents the expected accuracy, which is the proportion of correct recognitions that would be expected by chance. To calculate κ for intent recognition and entity recognition performance, we constructed confusion matrices for each participant, capturing their classification results in two scenarios. Each confusion matrix consists of predefined intent and entity categories that users might encounter during the dialogue, as listed in Table 6. These matrices provide a structured evaluation of the chatbot's performance, offering insights into its ability to accurately interpret and process user input in both scenarios.

The results in Table 7 indicate that our chatbot performed well in both scenarios. Specifically, entity recognition in both scenarios and intent recognition in the runtime scenario demonstrated almost perfect agreement ($\kappa > 0.81$), underscoring the chatbot's effectiveness in these areas. In contrast, intent recognition in the static scenario showed substantial agreement ($0.61 < \kappa < 0.81$), with a κ value of 0.7296 ($VAR = 0.028$). Through the experiment, we observed that users were more likely to inquire about various aspects of the product in the static scenario, which sometimes led to classification errors by the chatbot. Conversely, inquiries about tasks in the runtime scenario were relatively straightforward, contributing to higher accuracy in intent recognition. Overall, these findings indicate that the chatbot's effectiveness is shaped by the type and complexity of user inquiries in both static and runtime contexts. The almost perfect agreement in most categories highlights the system's robust understanding capabilities, particularly in runtime task-related interactions.

6.3.2 Dialogue Costs

The metrics depicted in Table 8 were adapted from the work of Hung et al. [57] and cover both objective and subjective aspects of dialogue evaluation. The first four metrics represent objective measures, while the remaining metrics are subjective.

To assess the subjective metrics (*Usability*, *Clarity*, *Naturalness*, *Friendliness*, *Robustness to misunderstandings*, and *Willingness to use the system again*)-we designed Questionnaire, 3 shown in Table 4, containing 18 questions derived from prior validated chatbot evaluation studies [58, 59]. Participants rated each item on a 7-point Likert scale [60], ranging from 1 ("strongly disagree") to 7 ("strongly agree").

As the subjective dimensions listed in Table 4 capture distinct aspects of dialogue quality, their relative contributions to overall evaluation cannot be assumed to be equal. Instead of relying on uniform weighting, we employed principal component analysis (PCA) [61] to the raw questionnaire responses, thereby extracting empirically grounded weights that reflect the underlying variance structure in participants' evaluations. This data-driven method allowed us to compute the relative weight of each item based on the variance structure in user ratings. Specifically, we used the loading coefficients from the first principal component (PC1), which captures the most variance and reflects the dominant factor in subjective dialogue evaluation. The resulting weights, normalised to sum to 1, are listed below:

Q1: 0.026, *Q2:* 0.0159, *Q3:* 0.060, *Q4:* 0.0885, *Q5:* 0.0819, *Q6:* 0.1115, *Q7:* 0.0256, *Q8:* 0.1431, *Q9:* 0.0109, *Q10:* 0.0492, *Q11:* 0.0280, *Q12:* 0.0336, *Q13:* 0.0652, *Q14:* 0.0144, *Q15:* 0.0765, *Q16:* 0.0984, *Q17:* 0.0100, *Q18:* 0.0612.

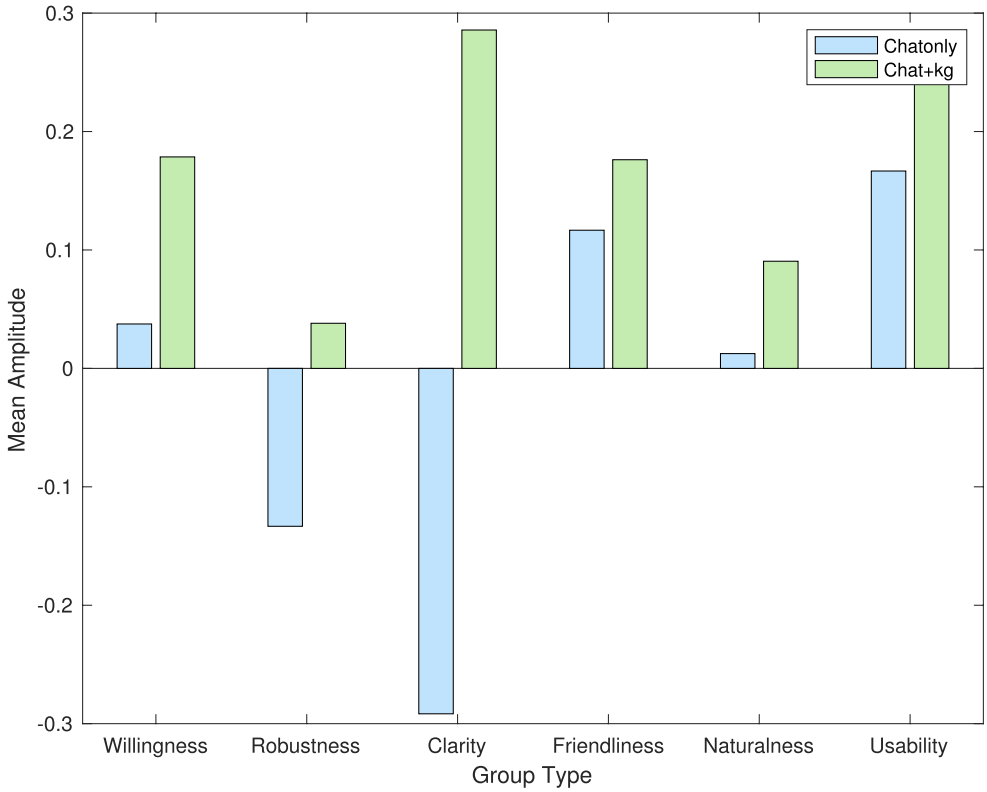


Fig. 12 Comparison of participants' subjective evaluation between the chatbot and Chatbot+KG group

Using these weights in a composite score ensures that dimensions with greater explanatory power (i.e., variance contribution) have a proportionally stronger impact on the final evaluation. To calculate the final subjective score of each participant, their responses were linearly mapped from the original 7-point scale to the interval $[-3, 3]$, after which the weighted sum was calculated using the PCA-derived weights. This mapping facilitated interpretable and consistent quantitative analysis across participants.

The scores for the six subjective metrics were based on the questionnaires filled out after the experiment, as shown in Fig. 12. Before analyzing the results, the Kolmogorov–Smirnov (KS) test [62] was implemented to evaluate the differences of the two groups. The test shows that the answers to the questionnaire come from different distributions with a p-value of 0.0186. Furthermore, the results of the Chatbot and Chatbot+KG groups were compared to better understand the role of the proposed KGs in effectively conveying scenario information to the users. The scores illustrate participants' subjective evaluation of the two interaction methods under each metric.

The results indicate a significant improvement in all evaluation metrics for the Chatbot+KG group compared to the Chatbot group. The introduction of KGs resulted in a positive shift in evaluations for both *Robustness to misunderstandings* and *Clarity*, which had previously received negative ratings. Notably, *Clarity* received the highest rating, underscoring the intuitive interpretability of KGs. These findings underscore the substantial benefits of integrating KGs with the chatbot, demonstrating marked improvements in user satisfaction across all evaluated metrics.

For the objective metrics of dialogue costs, including total *User turns*, total *Elapsed time per turn*, number of *Re-prompts*, and number of *Inappropriate responses* were calculated based on the experiment results. This objective evaluation focuses exclusively on the chatbot's performance, independent of the KGs, enabling a direct comparison between different scenarios. To ensure comparability across variables, all values were normalized using min-max normalization. The normalized results are presented in Fig. 13.

KS test was also implemented here for significance evaluation. The *Total user turns* decreased by 30.5% when transitioning from Scenario 1 to Scenario 2 within the same 5-minute duration. However, due to the large $VAR = 5.1841$, KS test gives $p = 0.14$, not showing significant differences. This result reflects the increased complexity of robotic tasks in runtime scenarios, making them more challenging for users to interpret and memorize. This complexity is further evidenced by the 1.8 times increase in the number of *Re-prompts* from Scenario 1 to Scenario 2 with significant differences ($p = 0.0468$), matching the results shown in Fig.

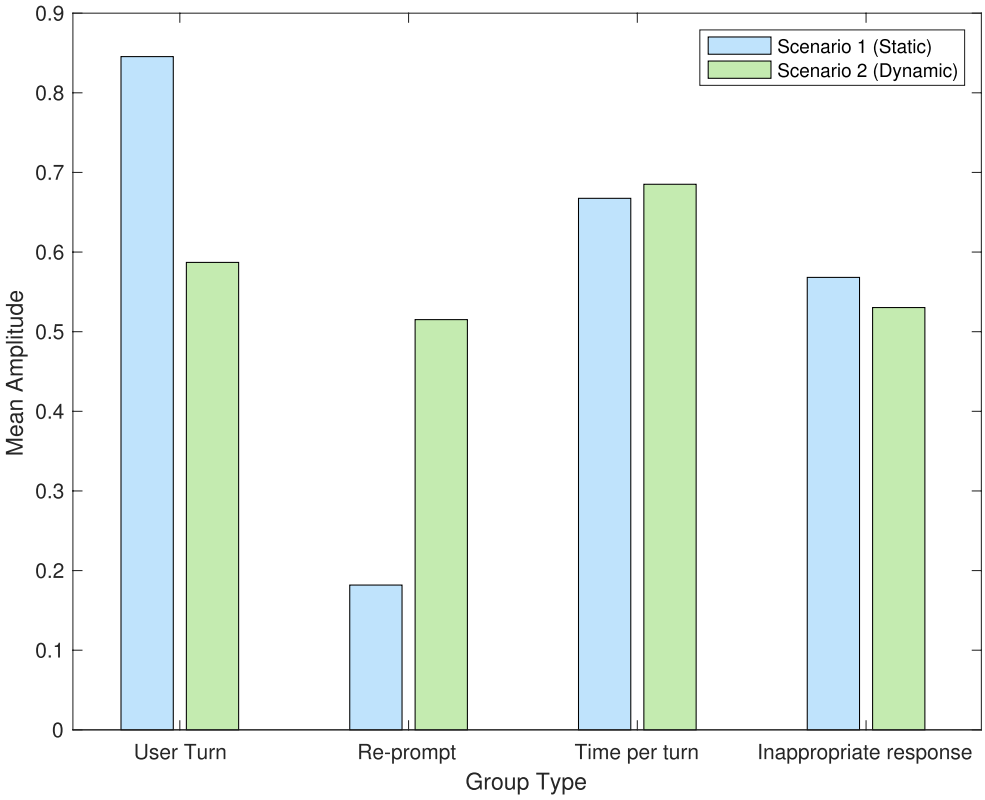


Fig. 13 Comparison of participants' objective evaluation between the Scenario 1 and Scenario 2

13. In contrast, the *Total elapsed time per turn* and the number of *Inappropriate responses* remained nearly constant, indicating that the chatbot maintained a consistent level of interaction quality regardless of task complexity or scenario type. The differences obtained from the KS-test between those two subgroups are not significant. These findings suggest that while the chatbot's interaction quality remains stable, the complexity of comprehending runtime information demands more user effort than static information.

6.3.3 Performance Score

The overall performance score was obtained using Eq. 4 from PARADISE. Here, P represents the subjective score derived from Questionnaire 3, described in the subsection 6.3.2. The coefficient k is derived from confusion matrices measuring intent and entity recognition accuracy. The terms $c_i, i \in [1:4]$ correspond to objective metrics of dialogue costs, normalized using the $\mathcal{N}$ Z-Score normalization function. To determine the weights α and w_i, individual participant scores P_i, $\mathcal{N}(k)$, and $\mathcal{N}(c_i)$ for each group were used in a regression fitting process as per Eq. 4. Once the regression weights were obtained, the final performance scores for each group were calculated by applying these weights to the mean Z-Scores of k and c_i.

$$P = (\alpha * \mathcal{N}(k)) - \sum_i w_i * \mathcal{N}(c_i), i \in [1:4] \tag{4}$$

The ANOVA results, as shown in Fig. 14, indicate that the Chatbot+KG group ($\overline{X} = 0.5866$, VAR=0.5722) significantly outperformed the Chatbot-only group ($\overline{X} = -1.0265$, VAR=0.4281; p=0.0047). The performance score for the Chatbot group was negative and clustered near "Slightly Disagree", reflecting limited user satisfaction. In contrast, the inclusion of KGs markedly elevated the performance score to a satisfactory range. Notably, although the Chatbot+KG group exhibited slightly greater variance, its entire score range (except for a single outlier) remained higher than that of the Chatbot group. This demonstrates the substantial benefits of KG integration in improving the quality of task-oriented interaction. The relatively large variances observed in both groups may be attributed to individual differences in user interaction behaviours.

7 Conclusion

In this paper, we developed a KG-based dialogue system that integrates structured knowledge representation with natural language interaction to facilitate scenario-specific

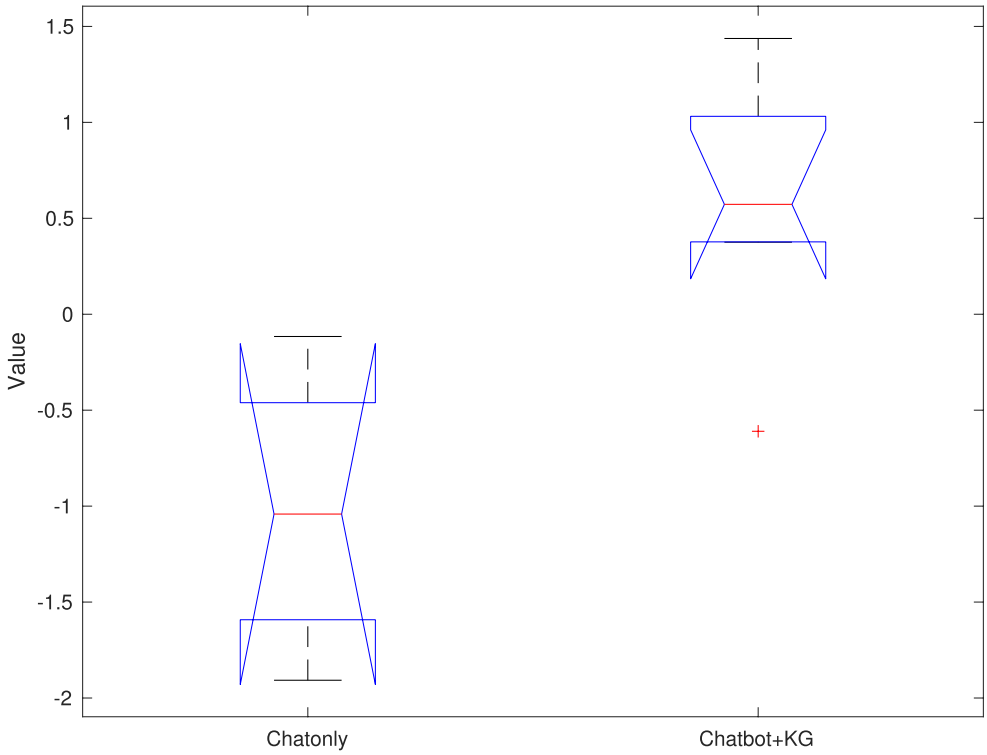


Fig. 14 Notched boxplot of final performance scores for the chatbot and Chatbot+KG groups. The y-axis represents composite scores mapped from the 7-point Likert scale. The red line marks the median, the notch its confidence interval, and the red cross an outlier. The Chatbot+KG achieved significantly higher scores than the chatbot-only group ($p = 0.0047$)

explanations in HRI. Using concepts from several IEEE standardized ontologies, we designed the IORA-LPG schema to efficiently represent robotic knowledge, demonstrated through a case study in the retail setting of AIRLab Delft. Unlike existing NLP-based dialogue systems primarily focused on robot command execution, the proposed KG-Retailbot is specifically designed to interpret and communicate structured scenario knowledge using KGs within a standardized robotic knowledge hierarchy.

A between-subject study with 25 participants was conducted to evaluate the effectiveness and performance of KG-Retailbot. Statistical analysis, including ANOVA, Tukey's HSD post hoc tests, and Kolmogorov-Smirnov tests, confirmed significant differences between experimental conditions, ensuring the robustness of our results. The results demonstrated that integrating KGs with a chatbot significantly enhanced users' ability to recall task-related information ($p < 0.05$) and improved product information retrieval accuracy compared to the Chatbot-only group. These results highlight the advantages of structured knowledge representation in facilitating information retrieval within HRI.

Beyond information retrieval accuracy, chatbot performance was evaluated using both objective and subjective metrics within the PARADISE framework. Subjective evaluations revealed that the Chatbot+KG system significantly improved user satisfaction, particularly in *clarity* and *Robustness to misunderstandings*. Objective measures showed that while the chatbot's interaction quality remained stable across scenarios, runtime tasks led to a significant increase in user re-prompts ($p < 0.05$), highlighting the added cognitive demand in understanding dynamic task information. The final performance score further confirmed the benefits of the KG integration, demonstrating its role in enhancing both task success and user experience.

In future work, improving bidirectional information exchange within this system could allow users to dynamically modify scenario data through input channels. Additionally, integrating a synchronized LLM+KG assistant could combine the generative capabilities of LLMs with the structured, domain-specific knowledge of KGs [63], enabling more flexible and context-aware explanations. Although our study used a default text-based CLI as a baseline for comparison, we acknowledge that enhanced CLI designs, incorporating structured command interfaces or visual elements, could provide more intuitive access to scenario information. Future work could explore these variations to provide a broader evaluation of interaction modalities in HRI. Finally, investigating the impact of human trust on system explainability across different interaction modalities could offer valuable insights into optimizing user engagement and system transparency.

Table

Table 9 Query templates for different users' intents

Action server	Cypher template
action_response_product_in_env	MATCH (o:{object_type} {attrs}) RETURN o
action_response_product_property action_response_specific_product	MATCH (o:Object) WHERE o.name=? RETURN properties(o)
action_response_product_location	MATCH (o:Object)-[r*2]->;(p:PositionRegion) WHERE o.name=?
action_response_furniture	MATCH (o:Object)-[r*2]->;(p:PositionRegion) WHERE p.value=?
action_response_specific_task	MATCH (t:Task)<-[r*2]-(a:Agent Plan) WHERE t.name=? RETURN t,a
action_response_current_task	MATCH (t:Task)<-[r*2]-(a:Agent Plan) Where t.status='ACTIVE'
action_response_previous_task	MATCH (t:Task)<-[r*2]-(a:Agent Plan) Where p.status='SUCCEEDED'
action_response_current_action	MATCH (a:Agent Plan)-[r*3]->;(s:Subtask) WHERE s.status='ACTIVE'

Acknowledgements The work was supported by AIRLab Delft. We extend our gratitude to the KAS Lab, TUDelft, for assistance with participant recruitment and insightful feedback on the experiment.

Funding The authors declare that financial support was partially received for the research, authorship, and/or publication of this article. The authors are grateful to the EPSRC doctoral scholarship, project reference 2849146, for the first author of this work.

Data Availability The datasets generated and/or analyzed during this study can be obtained from the corresponding author upon reasonable request. To ensure participant confidentiality and comply with ethical guidelines, access to the data is restricted and will be provided under specific conditions, ensuring it is used solely for research purposes.

Declarations

Ethics Approval The research protocols were thoroughly reviewed and approved by the Human Research Ethics Committee of Delft University of Technology. As part of the approval process, essential documents such as the data management plan, risk-planning document, and consent form were submitted and scrutinized.

Consent to Participate Informed consent was obtained from all participants prior to their inclusion in the study.

Conflict of Interest The authors declare that the research was conducted without any commercial or financial relationship that could be construed as a potential conflict of interest. The authors have no relevant financial or non-financial interests to disclose. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ehsan U, Liao QV, Muller M, Riedl MO, Weisz JD (2021) Expanding explainability: towards social transparency in AI systems. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp 1–19
2. Sado F, Loo CK, Liew WS, Kerzel M, Wermter S (2023) Explainable goal-driven agents and robots-a comprehensive review. ACM Comput Surv 55(10):1–41
3. Anjomshoae S, Najjar A, Calvaresi D, Främling K (2019) Explainable agents and robots: results from a systematic literature review. In 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019, International Foundation for Autonomous Agents and Multiagent Systems, Montreal, Canada, 1078–1088). May 13–17, 2019
4. Setchi R, Dehkordi MB, Khan JS (2020) Explainable robotics in human-robot interactions. Procedia Comput Sci 176:3057–3066
5. Pieters W (2011) Explanation and trust: What to tell the user in security and AI? Ethics Inf Technol 13(1):53–64
6. Marge M, Espy-Wilson C, Ward N (2020) Spoken language interaction with robots: research issues and recommendations, report from the nsf future directions workshop. arXiv preprint arXiv:2011.05533
7. He L, Lee K, Lewis M, Zettlemoyer L (2017) Deep semantic role labeling: What works and what's next. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 473–483
8. Lafferty J, McCallum A, Pereira FC (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data
9. Sukhwani M, Duggal V, Zahrai S (2021) Dynamic knowledge graphs as semantic memory model for industrial robots. arXiv preprint arXiv:2101.01099
10. Li Z, Mu Y, Sun Z, Song S, Su J, Zhang J (2021) Intention understanding in human–robot interaction based on visual-nlp semantics. Front Neurorob 14:610139
11. Koubaa A (2023) Rosgpt: next-generation human-robot interaction with chatgpt and ros
12. Benjdira B, Koubaa A, Ali AM (2023) Rosgpt_vision: commanding robots using only language models' prompts. arXiv preprint arXiv:2308.11236
13. Grassi L, Recchiuto CT, Sgorbissa A (2022) Knowledge triggering, extraction and storage via human–robot verbal interaction. Robot Auton Syst 148:103938
14. Fujii A, Kristiina J (2022) Open source system integration towards natural interaction with robots. In 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp 768–772. IEEE

15. Fujii A, Jokinen K, Okada K, Inaba M (2022) Development of dialogue system architecture toward co-creating social intelligence when talking with a partner robot. Front Robot AI 9:933001
16. Bocklisch T, Faulkner J, Pawlowski N, Nichol A (2017) Rasa: open source language understanding and dialogue management. arXiv preprint arXiv:1712.05181
17. Mohamed Y, Lemaignan S (2021) Ros for human-robot interaction. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3020–3027. IEEE
18. Lemaignan S, Cooper S, Ros R, Ferrini L, Andriella A, Irisarri A (2023) Open-source natural language processing on the pal robotics ari social robot. In Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction, pp 907–908
19. Jokinen K, Nishimura S, Watanabe K, Nishimura T (2019) Human-robot dialogues for explaining activities. In: 9th International workshop on spoken dialogue system technology. Springer, pp 239–251
20. Bruno B, Recchiuto CT, Papadopoulos I, Saffiotti A, Koulouglioti C, Menicatti R, Mastrogiovanni F, Zaccaria R, Sgorbissa A (2019) Knowledge representation for culturally competent personal robots: requirements, design principles, implementation, and assessment. Int J Soc Robot 11:515–538
21. Berners-Lee T (2006) Linked data. [EB/OL]. https://www.w3.org/DesignIssues/LinkedData.html
22. Ait-Mlouk A, Jiang L (2020) Kbot: a knowledge graph based chatbot for natural language understanding over linked data. IEEE Access 8:149220–149230
23. Robinson I, Webber J, Eifrem E (2015) Graph databases: new opportunities for connected data. "O'Reilly Media, Inc.
24. Wilcock G (2019) Citytalk: robots that talk to tourists and can switch domains during the dialogue. In: 9th International workshop on spoken dialogue system technology. Springer, pp 411–417
25. Wilcock G (2021) Recognising flexible intents and multiple domains in extended human-robot dialogues. In Annual Conference of the Japanese Society for Artificial Intelligence, Springer, pp 142–153
26. Al Moubayed S, Beskow J, Skantze G, Granström B (2012) Furhat: a back-projected human-like robot head for multiparty human-machine interaction. In: Cognitive behavioural systems: COST 2102 international training school. Dresden, Germany, February 21-26, 2011, Revised Selected Papers, pp 114–130. Springer
27. Wilcock G, Jokinen K (2022) Conversational AI and knowledge graphs for social robot interaction. In 2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp 1090–1094. IEEE
28. Hao X, Ji Z, Li X, Yin L, Liu L, Sun M, Liu Q, Yang R (2021) Construction and application of a knowledge graph. Remote Sens 13(13):2511
29. (2015) Ieee standard ontologies for robotics and automation. In IEEE Std 1872-2015:1–60. https://doi.org/10.1109/IEEESTD.2015.7084073
30. Pease A, Niles I, Li J (2002) The suggested upper merged ontology: a large ontology for the semantic web and its applications. In Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web, vol 28. pp 7–10
31. Fiorini SR, Carbonera JL, Gonçalves P, Jorge VA, Rey VF, Haidegger T, Abel M, Redfield SA, Balakirsky S, Ragavan V et al (2015) Extensions to the core ontology for robotics and automation. Robot Comput-Integr Manuf 33:3–11
32. Association IS et al (2017) P7007—ontological standard for ethically driven robotics and automation systems
33. Balakirsky S, Schlenoff C, Rama Fiorini S, Redfield S, Barreto M, Nakawala H, Carbonera JL, Soldatova L, Bermejo-Alonso J, Maikore F et al (2017) Towards a robot task ontology standard. In International Manufacturing Science and Engineering Conference, vol 50749. American Society of Mechanical Engineers, pp. 003–04049
34. Lassila O, Swick RR et al (1998) Resource description framework (rdf) model and syntax specification
35. McGuinness DL, Van Harmelen F et al (2004) Owl web ontology language overview. W3C Recommendation 10(10):2004
36. McBride B (2002) Jena: a semantic web toolkit. IEEE Internet Comput 6(6):55–59
37. Rodriguez MA, Neubauer P (2010) Constructions from dots and lines. arXiv preprint arXiv:1006.2361
38. Rodriguez MA, Neubauer P (2012) The graph traversal pattern. Graph Data Manag: Techniques Appl 29–46. IGI global
39. Alocci D, Mariethoz J, Horlacher O, Bolleman JT, Campbell MP, Lisacek F (2015) Property graph vs rdf triple store: a comparison on glycan substructure search. PLoS One 10(12):0144578
40. Barrasa J (2017) RDF triple stores vs. labeled property graphs. What's the difference? https://neo4j.com/blog/rdf-triple-store-vs-labeled-property-graph-difference/
41. Pezzato C, Hernández Corbato C, Bonhof S, Wisse M (2023) Active inference and behavior trees for reactive action planning and execution in robotics. IEEE Trans on Robot
42. Jordan G, Jordan G (2014) Neo4j + python. Practical Neo4j 169–213
43. Fiorini SR IEEE1872-owl - GitHub repository. https://github.com/srfiorini/IEEE1872-owl
44. Neo4j Labs neosemantics (n10s): Neo4j RDF & Semantics toolkit. https://neo4j.com/labs/neosemantics/
45. Honnibal M, Montani I (2017) Spacy 2: natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. To appear 7(1):411–420
46. Bunk T, Varshneya D, Vlasov V, Nichol A (2020) Diet: lightweight language understanding for dialogue systems. arXiv preprint arXiv:2004.09936
47. Vlasov V, Mosig JE, Nichol A (2019) Dialogue transformers. arXiv preprint arXiv:1910.00486
48. Francis N, Green A, Guagliardo P, Libkin L, Lindaaker T, Marsault V, Plantikow S, Rydberg M, Selmer P, Taylor A (2018) Cypher: an evolving query language for property graphs. In Proceedings of the 2018 International Conference on Management of Data, pp 1433–1445
49. Greenwald AG (1976) Within-subjects designs: to use or not to use? Psychological Bull 83(2):314
50. Bartko JJ (1966) The intraclass correlation coefficient as a measure of reliability. Psychological Rep 19(1):3–11
51. Walker MA, Litman DJ, Kamm CA, Abella A (1997) Paradise: a framework for evaluating spoken dialogue agents. https://doi.org/10.48550/ARXIV.CMP-LG/9704004
52. Shrout PE, Fleiss JL (1979) Intraclass correlations: uses in assessing rater reliability. Psychological Bull 86(2):420
53. St L, Wold S et al (1989) Analysis of variance (anova). Chemom Intell Lab Syst 6(4):259–272
54. Abdi H, Williams LJ (2010) Tukey's honestly significant difference (hsd) test. Encycl Res Des 3(1):1–5
55. Cohen J (1973) Eta-squared and partial eta-squared in fixed factor anova designs. Educ Psychological Meas 33(1):107–112
56. Carletta J (1996) Assessing agreement on classification tasks: the kappa statistic. arXiv preprint cmp-lg/9602004
57. Hung V, Elvir M, Gonzalez A, DeMara R (2009) Towards a method for evaluating naturalness in conversational dialog systems. In 2009 IEEE International Conference on Systems, Man and Cybernetics, pp 1236–1241. IEEE
58. Hone KS, Graham R (2001) Subjective assessment of speech-system interface usability. In Seventh European Conference on Speech Communication and Technology

59. Polkosky MD (2005) Toward a social-cognitive psychology of speech technology: affective responses to speech-based e-service
60. Likert R (1932) A technique for the measurement of attitudes. Archives of psychology
61. Jolliffe IT (2002) Principal component analysis for special types of data. Springer
62. Berger VW, Zhou Y (2014) Kolmogorov–smirnov test: overview. Wiley statsref: statistics reference online
63. Pan S, Luo L, Wang Y, Chen C, Wang J, Wu X (2024) Unifying large language models and knowledge graphs: a roadmap. IEEE Trans on Knowl Data Eng

Ke Xu is a PhD candidate at the University of Sheffield, supervised by Prof. Sanja Dogramadzi, within the Medical Medical Robotics Lab. Her doctoral research is funded by the EPSRC Trustworthy Autonomous Systems Node in Resilience project. She received her BSc in Mechatronics from Chongqing University (2020), and MSc in Robotics from Delft University of Technology (2023). Her research interests include multimodal human–robot interaction, trustworthy robot-assisted dressing, and large language models for robotics.

Sen Yuan was born in Shanxi province, China, in 1998. He received his Ph.D. degree from the Delft University of Technology, Delft, The Netherlands, in 2024, in electrical engineering. He is currently a Post-Doctoral Researcher with the Group of Microwave Sensing, Signals and Systems (MS3), Delft University of Technology. His research interests include SAR imaging, signal processing for radar system, and new scheme of radar system design. He was the recipient of the European Microwave Association Student Grant in 2021, 2022, 2023 and 2024. He serves as an Associate Editor for IEEE Aerospace and Electronic Systems Magazine and is an associate member of the IEEE Signal Processing Society Autonomous Systems Initiative (ASI).

Sanja Dogramadzi received the MEng degree in mechanical and control engineering from the University of Belgrade and a Ph.D. degree in robotics from the University of Newcastle, U.K. She is the Medical Robotics Group lead at the School of Electrical and Electronic Engineering, University of Sheffield. She has led many research projects funded by the U.K. Councils and European Commission and published over hundred peer reviewed papers and book chapters. Prof. Dogramadzi's current research interests include intelligent assistive robots for physical assistance based on safe and ethical HRI. She has supervised over 50 PhD and research master theses.

Carlos Hernandez Corbato is an Associate Professor at the Cognitive Robotics Department in the Faculty of Mechanical Engineering of TU Delft, The Netherlands. He participates in the EU projects CoreSense, METATOOL, and REMARO, has served as scientific coordinator in other EU projects, and won the Amazon Robotics Challenge 2016 with Team Delft. His research interests include software architectures for robotics, knowledge representation and reasoning, model-based systems engineering, and self-adaptive systems, and he teaches these topics in the MSc Robotics Program. Carlos holds MSc degrees in engineering (2006) and automation and robotics (2008), as well as a PhD (2013) from the Universidad Politécnica de Madrid.