PepMotion, your Personal cinema assistant

Pamfil Gabriel (1938844)
Palumbo Simone (1939378)
Panetta Francesco (1950468)

All students have equally contributed to the project

1 Introduction

1.1 Context and Motivation

The integration of social robots into public-facing environments has steadily progressed in recent years, particularly in service-oriented domains such as hospitality, retail, healthcare, and entertainment. Within this broader trajectory, the use of humanoid robots in cultural and recreational spaces such as museums, galleries, and cinemas has attracted increasing attention due to their potential to enhance user experience, streamline operations, and offer novel modes of interaction. Our project brings the Pepper humanoid robot into the context of a cinema, reimagining how people experience a movie outing. Far from being a sterile, lab-bound machine, Pepper is positioned as a friendly receptionist and assistant who greets guests upon arrival, offers personalized film recommendations based on past preferences, guides visitors to the correct screening rooms, and answers questions about the layout of the venue. This deployment isn't just about showcasing robotics technology; it's about human experience. Cinemas are often busy, dynamic environments. People arrive in groups, alone, with children, or in a rush. Some may be unfamiliar with the layout, hesitant to approach staff, or simply curious about what to watch. Pepper provides a non-judgmental, approachable source of guidance that responds with patience and consistency. On an economic level, the presence of a social robot like Pepper can also help ease operational pressures. By handling routine inquiries, directing foot traffic, and offering entertainment-related assistance, Pepper allows human staff to focus on more complex or personalized tasks. This redistribution of effort could lead to shorter wait times at counters, improved visitor flow, and higher satisfaction ratings. In the longer term, intelligent automation could help reduce staffing costs without compromising service quality. Moreover, by gathering usage data, Pepper could help management better understand visitor preferences and peak times, offering insights that could be used to optimize schedules, film programming, and customer engagement strategies. Ultimately, this project demonstrates how social robots like Pepper can be meaningfully

integrated into everyday cultural settings not as futuristic novelties, but as approachable, supportive agents that improve both user experience and operational efficiency. Pepper not only simplifies logistical aspects of a cinema visit but also helps create a more welcoming, inclusive atmosphere for a diverse range of visitors. While this implementation remains a prototype, it shows the broader potential of socially intelligent robots to reshape how we interact with public spaces making them more adaptive, empathetic, and human centered.

1.2 Objectives

The primary objective of this project is to develop and evaluate an autonomous socially intelligent robotic assistant capable of managing a multimodal HRI scenario within a cinema environment. Specifically, we aim to demonstrate Pepper's ability to:

- **User-adaptive dialogue management:** Pepper engages visitors through natural language dialogue or tablet interactions, answering questions related to current films and others that will be described later on, while adapting its tone based on the visitor's age group (e.g., more playful with children, more formal with adults).
- **Recommendation systems:** Pepper suggests movies tailored to the user's preferences, viewing history, and real-time availability. In addition to film recommendations, the system also proposes relevant concessions based on prior user behavior.
- **Navigation assistance:** With access to a spatial graph of the cinema's layout, Pepper offers clear, step-by-step guidance both verbally and through simple gestural cues to help visitors reach their destination efficiently.
- **Synchronized multimodal behavior:** Pepper synchronizes speech, tablet-based visuals and gesture animations to improve the clarity, comfort, and friendliness of the interaction.

Evaluation is conducted using several key HRI performance metrics, along with additional indicators:

- **Interaction fluency:** measured by system response time and conversation completion rate;
- **User satisfaction:** astandardized feedbacks on the interaction;
- **Navigation Capability:** evaluated by path smoothness and completion time;
- **Recommendation accuracy:** measures the relevance of Pepper's recommendations thanks to user's feedback.

By achieving these objectives, we aim to provide evidence that a social robot like Pepper can serve as a capable, adaptive, and socially aware assistant in public cultural spaces, demonstrating social intelligence and versatility.

1.3 Summary of the Results

In this project, we demonstrated that Pepper is capable of operating autonomously in a real-world cinema environment by interpreting user needs, reasoning over spatial and preference-based data, and delivering socially appropriate interactions. Starting from user identification via the tablet interface or speech, Pepper draws on a structured knowledge base consisting of a spatial graph of the venue and a dynamic user preference graph. This allows it to provide tailored movie recommendations, suggest appropriate concessions, and guide users efficiently to their destinations through synchronized verbal and gestural cues. The system adapts its communication style based on demographic inference (e.g., age), modulating tone and behavior accordingly. Multimodal interactions, delivered via speech, gestures, and tablet-based visuals, are orchestrated in real time to ensure clarity and user comfort, even in noisy or crowded settings. To improve over time, Pepper's knowledge base is updated continuously after each interaction. Feedback and session data are integrated into the recommendation graph, allowing the system to refine its suggestions and adapt to changing visitor behaviors and film schedules. Initial testing results show promising performance: a task completion rate of 85% and a Likert scale of 4.3 out of 5. Interaction fluency was rated positively across age groups, with minimal system response delays and high engagement in follow-up interactions. Navigation performance was assessed based on task completion and path smoothness. In that, the robot achieved a high completion rate, successfully guiding users to their destinations in all the trials. However, path smoothness was impacted by the discrete nature of the cinema map, which often resulted in sharp 90-degree turns at corridor intersections. While this did not prevent successful navigation, it occasionally led to less natural motion patterns, particularly in open or curved areas. Overall, the project offers both a technical contribution to the field of social robotics and a practical demonstration of how structured knowledge representations, adaptive learning, and multimodal communication can support efficient and user-friendly HRI in complex public spaces.

2 Related Work

The development of a socially interactive cinema assistant using the Pepper robot builds upon a rich body of research in Human-Robot Interaction (HRI), graph-based recommender systems, semantic environmental understanding, and benchmarking methodologies. In addition several university lectures and seminar presentations offered interesting insights that influenced the design and implementation of the system. Our primary research goal is to explore whether a mobile humanoid robot can guide users through a public venue while respect-

ing social and spatial conventions, provide personalized film recommendations through lightweight embedding graph-based models, and deliver a seamless multimodal interaction experience in real time. In contrast to traditional HRI implementations constrained to lab settings or narrow domains, this project integrates diverse capabilities into a functioning, end-to-end system deployed in a dynamic, public-facing environment. For robot interaction to be truly effective and natural, it is essential that the robot can perceive and interpret social signals, defined as short-term behavioral expressions of social and affective states. These signals include posture, facial expressions, non-verbal vocalizations, turn-taking patterns, and spatial proximity. Recognizing and responding to such cues allows the robot to modulate its behavior such as identifying confusion or disengagement and adjusting its speech tone or message content. One of the main ispiration was the course lecture on Social Signal Processing [5], it outlined a full pipeline for signal perception, interpretation, and generation. This framework is particularly relevant for a cinema assistant robot, where recognizing user emotions such as interest or satisfaction can enhance user experience by enabling adaptive communication strategies. Lastly, seminars on knowledge adaptation and passive learning [9] emphasized the value of continuous adaptation. This motivated our choice to implement a feedback-driven knowledge database, where each user's prior interactions and choices refine subsequent recommendations and behavioral strategies. The robot updates its internal model using minimal feedback such as prior film preferences or screen selections.

2.1 Human-Robot Interaction (HRI)

Our methodology is firmly grounded in the extensive body of research on Human-Robot Interaction (HRI), with a particular emphasis on socially-aware behaviors in public and semi-structured environments. One of the core principles of HRI is the consideration of proxemics and spatial behavior, as discussed by Joosse et al. [7], who proposes guidelines for adaptive spatial interaction between humans and robots. To enable natural and coherent multimodal interaction, we draw inspiration from the foundational work of Breazeal et al. [1] and the course lectures of our HRI course "Multi-Modal Interaction"[6], who emphasize the importance of synchronizing verbal and non-verbal communication. Specifically speech, gestures, gaze direction, and facial expression to facilitate smooth turn-taking and emotional resonance. Pepper's built-in capabilities, including a touchscreen interface, speech synthesis, and animated gestures, were orchestrated using these principles to minimize response latency and enhance interaction fluency. For evaluation, we adopt a performance framework that includes objective measures such as task success rate and interaction timing (i.e., latency between user input and robot response), as well as subjective metrics assessing perceived friendliness, helpfulness, and overall interaction quality. These criteria have been validated across numerous studies in public HRI deployments [12], and are particularly appropriate for assessing social robots operating in unstructured, real-world settings such as cinemas, museums, or shopping malls. These strategies are reinforced by our use of Pepper's native tablet and voice

interface, designed to minimize response latency and improve perceived fluency.

2.2 RBC

Our evaluation methodology is inspired by established benchmarking frameworks in service robotics, particularly those developed in the context of Robot Benchmarking Competition (RBC) [4]. These initiatives define evaluation protocols for robots operating in semi-structured human environments, with performance metrics spanning perception, task planning, action execution, and human interaction. Core evaluation metrics adopted in our project include task success rate, time-to-completion and user satisfaction that are measured through both objective and subjective channels. For the robot's guidance and navigation components, we integrate evaluation criteria inspired from socially-aware navigation literature [8]. These include:

- **Path smoothness:** evaluated via curvature and velocity profiles;
- **Proxemic compliance:** assessing the robot's ability to maintain socially acceptable distances during user accompaniment;
- **Perceived comfort:** obtained through post-interaction user questionnaires.

However, due to the limitations of the Coregraphe environment, specifically its inability to simulate dynamic crowds or unpredictable human motion, our current evaluation does not fully reflect the challenges of real-world cinema environments. A more robust simulation platform (e.g., ROS-based Gazebo with pedestrian models) would better replicate navigation under human congestion [11].

2.2.1 Graph-based Recommendation Systems

The film recommendation module in our system is built on a knowledge-graph-based approach leveraging recent advances in graph representation learning. While classical graph-based recommender systems such as PinSage [13] and LightGCN [3] use convolutional aggregation over user-item bipartite graphs, our system employs an embedding-based method tailored for knowledge graphs, specifically using RotatE [10]. Our graph database encodes users, movies, and their relationships—including interactions (e.g., views, likes), as well as semantic metadata such as genre, language and age rating, forming a heterogeneous multi-relational graph. Each entity and relation in the graph is embedded using the RotatE model. This capability is particularly useful for modeling directional user preferences or conditional genre affinities. As users interact with the system, feedback such as acceptance or rejection of movie suggestions is used to incrementally update the underlying graph and refine embeddings. To

evaluate and train our recommendation model, we employed a curated subset of the widely recognized MovieLens 1M dataset [2], a benchmark collection extensively used in the research community for collaborative filtering and content-based recommendation tasks. The original dataset contains one million ratings across approximately 6,000 users and 4,000 movies. However, to ensure compatibility with the computational and memory constraints of real-time inference on the Pepper robot, we extracted a domain-specific subset focused on movies that were seen by the majority of users. We use both objective and subjective metrics that are described in the successive section. In addition, we incorporate user feedback collected after each recommendation to iteratively refine the graph structure. This enables real-time personalization and improves the long-term relevance of the recommendations, following practices in adaptive recommendation benchmarking [**?**].

2.2.2 Summary of Contributions Beyond the State of the Art

While prior research in Human-Robot Interaction has explored individual components such as personalized recommendation [10, 3], socially-aware navigation [8], or multimodal dialogue [6, 1], these capabilities are often studied in isolation or under controlled laboratory conditions. Our work advances the state of the art by integrating these dimensions into a unified system, specifically designed for real-world deployment in public entertainment venues such as cinemas. Key contributions include:

- **Unified HRI framework:** We combine personalized content recommendation, proxemically aware navigation, and socially competent interaction into a cohesive user experience, enabling seamless transitions between guidance and conversation.
- **Knowledge-graph-enhanced personalization:** Our recommendation system leverages a multi-relational graph model based on RotatE embeddings [10], enabling fine-grained user modeling and context-sensitive suggestions grounded in cinema-specific metadata.
- **Multimodal and socially responsive interface:** Drawing from social signal processing [5] and multimodal interaction templates [6], our system adapts verbal and non-verbal output (e.g., gestures, distance regulation, tablet-based content) to the user profile and situational context.
- **Operational deployment in constrained environments:** Unlike works based purely on simulation or static datasets, our system runs on a Pepper robot within a live demonstrator, integrating the NAOqi stack, Coregraphe routines, and online learning loops for personalization.

By orchestrating these components, we move toward socially intelligent robotic assistants capable of operating autonomously in semi-structured, human-populated environments. This system represents a step forward in realizing long-term, human-centered service robots for public spaces.

3 Integrated Solution

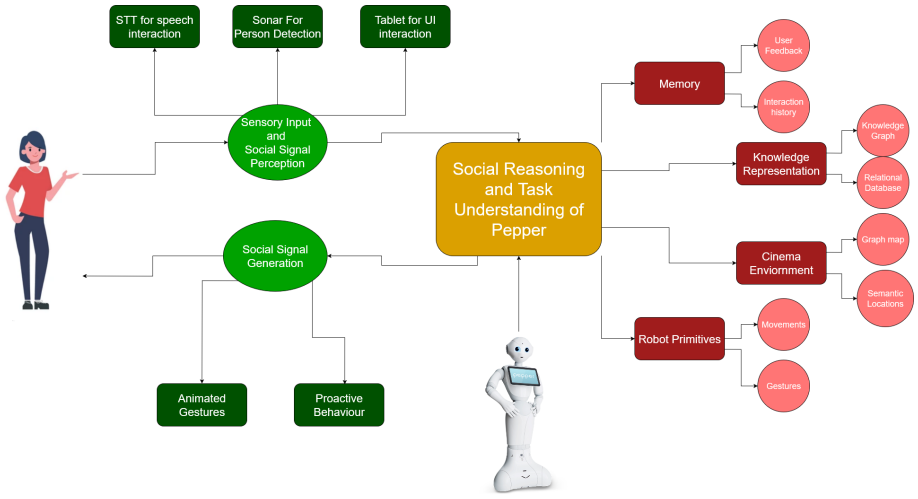


Figure 1: Our HRI Architecture

The workflow illustrated in Figure 1 illustrates all the modules integrated into our framework and how they interact with each other. The first stage of the Human-Robot Interaction involves detecting the user and initiating communication through social signals. To this end, the system employs several input modalities: speech-to-text (STT), a tablet-based UI, and sonar-based person detection. While in simulation (e.g., on Choregraphe) certain aspects—such as real person detection—are hardcoded, in real-world deployment they operate effectively. To express social signals, the Pepper robot adopts a proactive behavior, including animated gestures. In the Knowledge-Driven Modules, we implemented a persistent memory that records user interactions and feedback. For knowledge representation, the system combines a relational database with a knowledge graph that supports personalized recommendations. Specifically, the graph stores data on users' movie preferences and leverages this history to tailor suggestions accordingly. An additional component is the cinema map: a graph-based model representing key landmarks within the cinema, such as the box office, bar, restrooms, and each screening room. This map enables the robot to provide guided navigation to users when needed. Finally, the robot includes essential capabilities for locomotion and gesture execution, which are crucial for both navigation and delivering a natural, engaging user experience.

3.1 Human-Robot Interaction (HRI)

Interaction Modalities

The project utilizes a multimodal approach for both human-to-robot and robot-to-human communication. In fact the user can communicate either with its voice

which serves as the primary input modality, allowing users to provide information, answer questions, and make requests through natural language utterances, or by touching the tablet, in which case the interaction will follow the same conversational flow. Users can interact via buttons for movie selections, concessions, and other actions, supporting a multimodal input strategy. On the other hand, language is the robot's main output, which is supported by visuals on the tablet, like images, text, and buttons, to make things clearer. This isn't just useful in noisy environments or when the user is distracted; in many cases, visuals are simply more effective than speech alone. For example, when giving directions, showing a map on the screen makes the instruction much easier to understand than describing it verbally.

Social Signals

The robot conveys a range of non-verbal social cues using animations that express emotions and attitudes such as enthusiasm, thoughtfulness, confusion, or agreement. These cues are synchronized with dialogue and designed to reflect natural human behavior, enhancing the social presence and relatability of the robot during interaction.

Memory and Background Knowledge

The system integrates persistent interaction data, such as bookings, showtimes and logged-in users, into a relational database, alongside movie information from the MovieLens dataset, which contains 1 million ratings from 6,000 users on 4,000 movies. This dataset serves as the foundation for the recommendation system. The robot possesses also knowledge of cinema locations (e.g., screen numbers, box office, concession stand and restroom) to provide accurate directions. The recommendation system determines its approach based on the user's activity within the database. If the user has liked a sufficient number of movies, the system uses the RotatE model to make reliable predictions. In this case, it infers the most likely movies the user would enjoy by analyzing node embeddings in a knowledge graph. This knowledge graph includes nodes representing movies, users, genres, and age groups. The relationships between them are defined as likes, is_genre, and has_age. If there isn't enough user data for a robust prediction, the system instead recommends movies based on the user's stated preferences for genre and age group.

Human Mental Models

The robot is designed to follow various human mental models, adapting its language and tone according to the user's declared age, whether a child, teen, or adult, to match different cognitive and social expectations. It recommends movies based on a combination of past feedback and stated preferences, showing an understanding that individual tastes and viewing history can help predict future interests. The interaction follows familiar patterns of human conversation, using turn-taking and expected responses to create a more natural flow. The robot also recognizes typical tasks associated with the goal of finding a film, buying snacks, or getting directions—and helps users accomplish these goals, reflecting an awareness of their practical needs and intentions.

Social Reasoning and Intelligence

The Social Reasoning module demonstrates social understanding by generat-

ing varied responses based on whether a liked movie matched the user's usual genre, indicating an awareness of consistency or novelty in preferences, and also shows temporal awareness, since by comparing the current time it determines if a movie is "upcoming" or if feedback should be requested for a previous watched movie. To maintain user engagement and avoid redundancy, it explicitly avoids recommending movies that the user has already watched, ensuring that suggestions remain novel. Moreover the robot offers assistance (guiding to screens, concession stands) and seeks confirmation for user intent, exhibiting supportive and helpful social behavior. The robot also demonstrates proactive social behavior, such as:

- Reminding users about booked showtimes.
- Suggests rating watched movies if not yet done.
- Offering to guide users to specific screen after booking tickets
- Proposing relevant snacks based on previous preferences.

3.1.1 Robot Behavior in Various Situations

Situation 1: A child approaches the robot for the first time.
Robot Behavior:

- *"Hey there! Welcome to our cozy cinema, glad to have you!"*
- *"Nice to meet you, [Child's Name]! Let's get to know you better. How would you describe your age group? Child, teen, adult?"*
- Upon receiving *"child"* as age: *"Wanna use the tablet to chat? It's super easy!"*
- After booking tickets: *"All set! Ask your parents to grab the tickets at the box office—it's on the tablet. Let me know when you're done!"* (robot indicates the tablet)
- Guiding to screen: *"Yay! Follow me!"*

Situation 2: An adult has just arrived at the cinema but the last time he didn't rate the movie he had seen, which has a different genre from his favourite one.
Robot Behavior:

- After greeting and welcoming back the user: *"You saw [Last Movie Watched] last time. Did you enjoy the film?"*
- If the user says *"no"*, the robot runs an embarrassed animation, conveying sympathy and says:

"I see. That's understandable, since it's a bit different from the [Preferred Genre] films you usually prefer. Thanks for the feedback!"

- *"Since you didn't like [Last Movie Watched], here are one or more movies for you that are playing today. Hurry up and book the tickets!"* (The robot immediately offers alternative recommendations after updating the negative feedback in the recommendation model.)

Situation 3: A returning teen customer really likes a certain snack.
Robot Behavior:

- *"Yo [Teen's Name], good to see you again!"*
- If the teen asks for concessions: *"I know you really like [Preferred Snack]. Would you like to add it to cart?"* (This demonstrates memory of past preferences.)

Situation 4: A teen has an upcoming movie booked and arrives at the cinema.
Robot Behavior:

- Upon detecting the teen's arrival:

  "Yo [Teen's Name], good to see you again!"
  "Heads up! Your movie [Movie Title] starts in just 10 minutes! Want me to guide you to Screen [Screen Number]?"

- If the teen says *"yes"*:

  "Cool, follow me!"

- (Robot initiates navigation to the screen.)

3.2 RBC

3.2.1 Functionality Benchmarks

The functionality benchmarking aims to evaluate key system components critical for robust Human-Robot Interaction (HRI), independently of the robot's overall task (e.g., recommending films or guiding users). The selected functionalities for evaluation are: Interaction fluency, Navigation and guidance, User satisfaction and recommendation inference.

Interaction Fluency Refers to Pepper's ability to sustain coherent and responsive conversations with users across multiple input/output modalities, primarily via speech and tablet interaction.
Metrics:

- *Turn-Taking Latency*: Average time between the end of a user's input and the start of the robot's response.
- *Interaction Completion Rate*: Fraction of conversations that successfully reach a defined task goal (e.g., booking a ticket or receiving a recommendation) without user abandonment or interruption.

How data is gathered:

- Interaction logs are collected from scripted test sessions involving both voice and tablet inputs across various user profiles (e.g., child, teen, adult).
- Timings are extracted from NAOqi dialogue logs and system event timestamps.
- Each conversation is annotated for success/failure and modality transitions to compute completion and recovery metrics.

User Satisfaction Captures the subjective evaluation of the interaction experience from the perspective of the human user. While qualitative, it serves as a critical indicator of the system's social acceptability, usability, and perceived intelligence, especially in public-facing deployments such as cinema assistance.
Metrics:

- *Likert-Scale Ratings*: Users rate their experience on standardized 5-point or 7-point Likert scales across dimensions such as friendliness, helpfulness, clarity, and enjoyment.
- *System Usability Scale (SUS)*: A 10-item standardized questionnaire used to assess perceived usability on a 0–100 scale.
- *Net Promoter Score (NPS)*: Single-question indicator asking whether the user would recommend the robot to others, yielding a promoter/detractor ratio.

How data is gathered:

- After completing an interaction (e.g., booking a movie or requesting guidance), users are asked to complete a brief post-interaction questionnaire.

Navigation and Guidance capabilities Pepper guides users to predefined destinations (e.g., screens, box office, concession stands) based on cinema layout knowledge.
Metrics:

- *Path Smoothness*: Qualitative score based on trajectory curvature and stop/start behavior.
- *Completion Time*: Time taken to reach the target location from the starting point.

How data is gathered:

- Simulated runs are performed in Choregraphe.
- Distance metrics and route completion times are extracted from the motion logs.

Recommendation Inference The recommendation module suggests personalized movies using knowledge graph embeddings (RotatE) trained on a subset of the MovieLens 1M dataset.
Metrics:

- *Mean Reciprocal Rank (MRR)*:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

- *Hits@10*:

$$\text{Hits@10} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \mathbb{I}(\text{rank}_i \leq 10)$$

  where Q is the set of test queries and rank_i is the position of the first correct result.

How data is gathered:

- Recommendations are generated in real-time using a local database on Pepper.
- Evaluation is based on held-out ratings from MovieLens

Note: Due to the limitations of the Choregraphe simulation environment, some dynamic behaviors (e.g., obstacle avoidance or crowd handling) could not be benchmarked accurately. These could be assessed in future deployments using more advanced simulation platforms (e.g., Gazebo).

3.2.2 Task-Level Benchmarks

To validate the system as an integrated, user-facing assistant, we define a set of task-level benchmarks aligned with typical interaction goals in a cinema setting. Each benchmark includes success metrics, participant involvement, variability across trials, and criteria for partial achievements.
User Identification and Preference Registration
Task-Level Metrics:

- Total time to complete User identification and Preference registration.
- Scoring:
  - +10 points if the user is identified in under than 1 minute.
  - +5 points if the user is identified in under than 1 minute and 30 seconds.
  - -2.5 points if user fails to be identified or abandons interaction.

Human Participants:

- Each trial involves a participant (either known to the system or interacting for the first time) approaching Pepper and initiating a session.
- Users are prompted to select their identity or declare new information using either tablet input or spoken commands.
- The participant is then asked to confirm age group and preferred genres.

Variability Across Trials:

- Response time and success may vary due to:
  - Differences in user familiarity with the tablet UI or voice interface.
  - User hesitation or indecision in selecting preferences.
- Returning users may complete this task faster due to saved profiles, while new users typically require additional time for registration.

Partial Achievements:

- If user is not recognized but completes preference selection and login manually, assign **+2.5 points** for partial recovery.
- If user provides age or genre only (not both), assign **+2.5 points**.

Movie Recommendation & Ticket Booking
Task-Level Metrics:

- Total time to complete recommendation and booking sequence via tablet or voice.
- Scoring:
  - +10 points if a recommended movie is accepted and booking is completed within 2 minutes.
  - +5 points if recommendation is accepted but booking is delayed beyond 2 minutes.
  - -2.5 points if the user reviews suggestions but does not make a selection.

Human Participants:

- Users interact with Pepper to receive personalized movie suggestions and finalize a booking via tablet.

Variability Across Trials:

- User familiarity with the robot interface can impact decision time.
- Different user profiles (e.g., child vs. adult) exhibit different response behaviors and decision confidence.

Partial Achievements:

- If a relevant recommendation is acknowledged but booking is skipped, assign -2.5 points.
- If the user navigates options but exits the session early, score -10 and label as "incomplete interaction."

Navigation to Target Location (Screen, Concession, Restroom)
Task-Level Metrics:

- Task completion time (robot reaches destination and stops appropriately).
- Scoring:
  - +10 points if destination is reached without interruption or replanning.
  - +5 points if the route is completed with minor clarification (e.g., user confirms destination mid-task).
  - 0 points if the robot reaches the wrong destination.

Human Participants:

- Users are asked to follow Pepper to a known target location within the simulated or real cinema map.

Variability Across Trials:

- User walking speed and path compliance may vary.
- Proximity to crowds or narrow passages may trigger obstacle-avoidance delays.

Partial Achievements:

- If navigation is paused and resumed due to user deviation or map error, assign -2.5 points.
- If Pepper requires manual redirection, mark as "assisted success" and score -5.

Concessions Interaction (Snack Ordering)
Task-Level Metrics:

- Successful retrieval of snack preferences and confirmation of order via tablet.
- Scoring:
  - +10 points if the robot recalls past preferences and receives a confirmed order within 1 minute.
  - +5 points if a new order is placed without preference recall.
  - -5 points if an order attempt is made but not completed.

Human Participants:

- Returning users interact with Pepper to purchase snacks, leveraging memory-based personalization.

Variability Across Trials:

- Snack availability or user decision fatigue may impact task length.
- Different age groups may engage more or less with the tablet-based interface.

Partial Achievements:

- If Pepper recalls a preference but the user chooses differently, assign +2.5 points.
- If order is canceled after adding something to cart, score -2.5.

3.2.3 Handling World Change and Uncertainty

Our architecture is inherently designed to operate in a dynamic and partially observable environment: a real cinema space where the film offering, user base, and spatial conditions can change continuously. The most prominent source of world change is the cinema schedule itself. Movies rotate weekly, and new titles replace old ones. To reflect this evolving world state, our system periodically updates the underlying data structures, specifically the film catalog stored in the knowledge graph and used by the recommendation module. This ensures that suggestions provided by the robot remain relevant to the current offerings. Beyond these scheduled updates, the system also deals with uncertainty in several other ways:

- **Partial user profiles**: When the system lacks sufficient information about a user (e.g., in the case of first-time visitors), it falls back on general heuristics, such as age-based or genre-based recommendations.

- **Recommendation confidence**: Each film suggestion is associated with a confidence score, computed based on how well it matches known user preferences. Low-confidence suggestions are presented with hedging language (e.g., "You might enjoy this"), while high-confidence ones are proposed more assertively.
- **Environment awareness**: While Pepper does not analyze physical ambient conditions (e.g., lighting), it does consider *social context* and *spatial layout*. For example, it infers whether part of the venue is temporarily inaccessible (e.g., bathroom), this state is reflected in the navigation graph, and the Pepper's response is updated accordingly.

Example: Weekly change in movie availability and its impact on recommendations.

- On Monday, the robot has a conversation with a returning user, who previously watched and rated multiple action movies. Based on her preferences, the system recommends *Fast and Furious 9*, currently showing in room 7.
- The following week, *Fast and Furious 9* is removed from the catalog and replaced by new titles. The system automatically updates its catalog accordingly.
- When the user returns and asks for a recommendation, the system detects that his previous favorite is no longer available. It retrieves the updated catalog and suggests a new available action movie, *John Wick 4*.

Example: A user asks for directions to the bathroom shortly after another user has already asked and gone there.

- After the user has asked directions for bathroom, the robot indicates the next node on the graph map to be reached and shows the map on the tablet
- At the next interaction the new user asks for directions for the bathroom, but this time the robot after pointing and showing the tablet says *"Unfortunately the bathroom is temporarily busy. Maybe try later? "* and shows the tablet location in red.

3.2.4 Integration with HRI Objectives

Each module contributes directly to enhancing human-robot cooperation, user understanding, and trust:

- **Speech and gesture modules** enable natural, accessible interaction across user age groups and noise conditions.
- **The knowledge graph** ensures contextual memory and personalization, improving user engagement and continuity.

- **The recommendation system** adapts over time, demonstrating attentiveness and user-specific insight.
- **Temporal reasoning and feedback prompts** (e.g., reminders to rate watched movies or alerting about showtimes) reflect social awareness and care.

Together, these elements form a cohesive, socially intelligent system capable of adapting to diverse users, environments, and goals. Future evaluations will expand to dynamic crowd conditions and unstructured environments to validate robustness in more realistic deployments.

4 Implementation and results

The Cinema Assistant was implemented as a modular system deployed on the Pepper humanoid robot. Our design follows a service-oriented architecture, where each functional component is encapsulated in an independent Python module, communicating via NAOqi APIs, shared memory signals, and WebSocket connections. The architecture reflects key requirements of real-world human-robot interaction (HRI) systems: robustness to failure, adaptability to dynamic environments, and clear modular separation between perception, reasoning, and interaction layers. To facilitate development and testing, the system was simulated extensively in Choregraphe, the official programming and simulation environment for Pepper.

4.1 Tools and Libraries Used

- **NAOqi SDK (Python 2.7):** Provides a comprehensive set of APIs for managing speech synthesis and recognition, motion control, dialog management, memory events, and other core robot functionalities.
- **SQLite3:** Used as an embedded database engine for efficient storage and querying of film metadata, enabling quick access to movie information during interaction.
- **MODIM Framework:** Powers the interactive graphical user interface on Pepper's tablet, utilizing web technologies such as HTML, JavaScript, and WebSocket communication to provide a smooth and responsive user experience.
- **Matplotlib:** Employed to generate the cinema map visualization, enabling the robot to display movie-related spatial information graphically.
- **TensorFlow:** Used to train the recommendation model, allowing the system to learn and improve movie suggestions based on user preferences.
- **Python Standard Libraries:** Including `json` for data serialization and `datetime` for handling temporal information, facilitating data exchange and event timing.

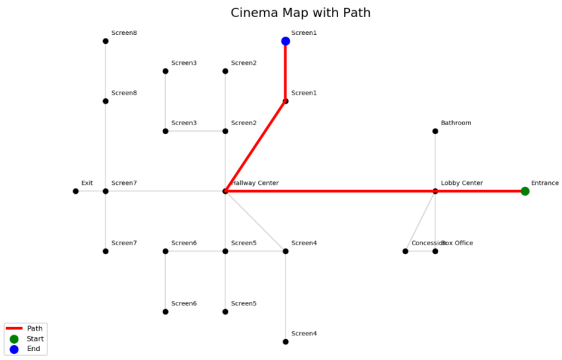
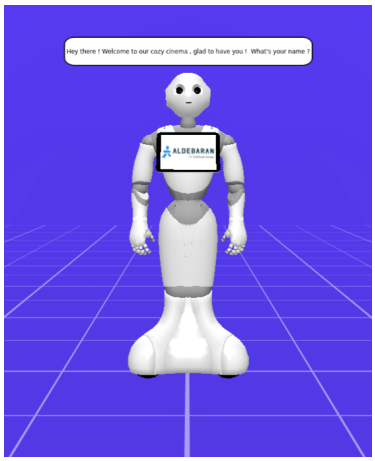


Figure 2: Example of planning on our Cinema Map

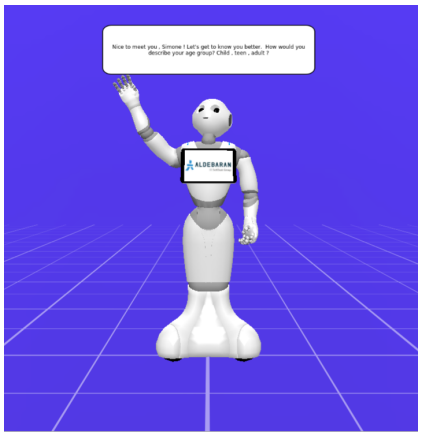
4.2 Cinema Environment and Motion Manager

The Cinema Environment module provides a graph-based representation of the cinema's physical layout. Important locations such as the entrance, lobby, box office, concession stand, bathrooms, hallways, and individual screening rooms are modeled as nodes within a coordinate system. Edges between nodes represent walkable paths, enabling pathfinding and navigation reasoning. This spatial model is implemented in the `CinemaMap` class, which supports computation of Euclidean distances and shortest paths using Dijkstra's algorithm. The map can be visualized with `matplotlib`, generating graphical representations highlighting the robot's current position, destination, and computed route. These visuals are saved as transparent PNG images for display on the robot's tablet interface. Due to simulation limitations within Pepper's Choregraphe environment, this map visualization is not directly available in the simulator but operates on the deployed robot system. To execute navigation and pointing gestures, the system leverages `ALMotion`, Pepper's native motion control module, through the `MotionManager` class. This class uses the motion proxy to perform physical movements such as:

- **Pointing and verbal direction**: Given a target location, the robot calculates the angle towards the next node in the shortest path and uses `ALMotion.moveTo` to orient itself accordingly. It performs arm movements by interpolating joint angles to simulate pointing gestures synchronized with verbal instructions.
- **Guided navigation**: The robot follows the computed shortest path node-by-node, turning towards each next point and moving forward using relative motions. Orientation is tracked and updated to ensure smooth, accurate navigation.
- **Customer engagement**: After guiding the user, the robot turns to face the customer and returns its head and arms to a neutral, attentive posture.



(a) Pepper greeting a customer



(b) Pepper asking a customer his age

Figure 3: Some interaction with pepper

4.3 Dialogue and Tablet Interaction Manager

The Dialogue and Tablet Interaction Manager is implemented as a modular system coordinating the robot's conversational abilities and the tablet's graphical user interface. It leverages a rule-based dialogue framework structured through semantic concepts and hierarchical proposals, which define dialogue states activated by recognized user inputs or memory events. The core components include:

- **ALDialog**, responsible for managing spoken dialogue, including turn-taking and speech recognition event handling, which trigger transitions between dialogue states based on user input.
- **ALMemory**, acting as an event hub where user inputs, captured via speech or tablet touch, are published as memory events. These events are continuously monitored by the assistant service to synchronize dialogue flow and GUI updates.
- The **MODIM framework**, providing a flexible GUI on Pepper's tablet. The GUI is developed using standard web technologies (HTML, JavaScript, CSS) and communicates with the dialogue manager via WebSocket.
- **ALAnimationPlayer**, which coordinates gestures and affective NAOqi animations synchronized with the dialogue to enhance expressivity and naturalness.

Dialogue management follows a hierarchical, event-driven approach, where user inputs matched against predefined semantic concepts trigger proposals that

manage interaction segments such as acquiring user identity, registering preferences, offering movie recommendations, or gathering feedback. The system maintains internal state variables within robot memory to adapt responses dynamically based on user preferences and contextual information.

4.4 Relational Database

To support dynamic interaction and personalized recommendation for new users, the system integrates a lightweight relational database using `SQLite3`. The database serves as the persistent storage backend for user profiles, film metadata, and interaction logs. The database schema includes the following key tables:

- `customers`: Stores unique user profiles, including identifiers, age group, and preference genre.
- `movies`: Contains metadata for available films, such as title, genre, duration, and screening room.
- `showtimes`: Defines scheduled screening times for each movie in specific rooms, enabling time-aware recommendation and navigation.
- `concessions`: Stores the list of available snacks and drinks at the cinema, which the robot can suggest during interaction.
- `bookings`: Logs reservations made by users for specific showtimes, including seat and room information.
- `orders`: Records snack/drink orders made by the user during interaction, linked to the selected items and user profile. It is used in order to recommend the preferred food.

4.5 Knowledge Graph

The knowledge graph (KG) is constructed as a structured representation of user preferences and item metadata, serving as the foundation for embedding-based learning using the RotatE model.

Data Processing and Triple Generation

The KG construction begins with parsing three data sources: user profiles, movie metadata, and user-item interactions. Each user is associated with demographic attributes such as age, which are discretized into categories (e.g., `child`, `teen`, `adult`) and transformed into triples of the form `(user, has_age, age_group)`. Similarly, each movie is linked with its genre information, forming triples like `(movie, is_genre, genre_label)`. User-item interactions are encoded by extracting positive ratings (rating ≥ 4) and converting them into `likes` relations, i.e., `(user, likes, movie)`. A subset of these interactions for training is selected, given the computational limitations of the robot.

Negative Sampling and Batching

For training purposes, negative samples are synthetically generated by corrupting either the head or tail of each positive triple. These are then combined with the original triples and labeled appropriately (positive or negative). Batches of training data are created through a shuffling and slicing mechanism, allowing for efficient gradient descent updates.

Integration with Embedding Model

The KG triples serve as input for the RotatE embedding model, where each entity is represented as a complex-valued vector, and each relation corresponds to a phase rotation in complex space. During training, the model optimizes the scoring function to distinguish valid triples from corrupted ones using a margin-based loss function. Let each training triple be of the form (h_i, r_i, t_i) with label $y_i \in \{+1, -1\}$, where $+1$ denotes a positive triple and -1 a negative one. The RotatE model learns complex embeddings and scores triples based on rotational distance in complex space. Each entity and relation is embedded as:

- Entity h, t have embeddings $\boldsymbol{h} = \boldsymbol{h}_{\text{re}} + i\boldsymbol{h}_{\text{im}}$, $\boldsymbol{t} = \boldsymbol{t}_{\text{re}} + i\boldsymbol{t}_{\text{im}}$
- Relation r is represented by a phase vector $\boldsymbol{\theta}_r$

We define:

$$\boldsymbol{r}_{\text{re}} = \cos\left(\boldsymbol{\theta}_r\right), \quad \boldsymbol{r}_{\text{im}} = \sin\left(\boldsymbol{\theta}_r\right)$$

The rotated head entity is computed as:

$$\text{Re}(\boldsymbol{h} \circ \boldsymbol{r}) = \boldsymbol{h}_{\text{re}} \circ \boldsymbol{r}_{\text{re}} - \boldsymbol{h}_{\text{im}} \circ \boldsymbol{r}_{\text{im}}$$

$$\text{Im}(\boldsymbol{h} \circ \boldsymbol{r}) = \boldsymbol{h}_{\text{re}} \circ \boldsymbol{r}_{\text{im}} + \boldsymbol{h}_{\text{im}} \circ \boldsymbol{r}_{\text{re}}$$

The score of a triple (h, r, t) is:

$$f(h, r, t) = \gamma - \|\boldsymbol{h} \circ \boldsymbol{r} - \boldsymbol{t}\|_1$$

That is,

$$f(h, r, t) = \gamma - \sum_{j=1}^{d} |\text{Re}(\boldsymbol{h} \circ \boldsymbol{r})_j - \boldsymbol{t}_{\text{re},j}| + |\text{Im}(\boldsymbol{h} \circ \boldsymbol{r})_j - \boldsymbol{t}_{\text{im},j}|$$

The loss is computed using the Softplus function:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \cdot f(h_i, r_i, t_i)\right)\right)$$

5 Results

The experimental evaluation of our Pepper-based cinema assistant demonstrates the system's ability to engage in socially intelligent, multimodal, and personalized human-robot interaction (HRI) in a public entertainment context. This section summarizes both functional and task-level benchmarks, and discusses interaction dynamics through nominal and non-nominal execution threads.

5.1 HRI – Example Execution Threads

Nominal Situation – Successful Booking and Navigation
A returning teen user approaches Pepper, who greets him by name using prior information stored in the local database. The robot offers a movie recommendation based on the user's feedback history and current preferences. The teen accepts the suggestion via tablet, completes booking, and accepts to be guided to the screening room. After watching the movie, the user accepts to rate the movie by giving a positive review.
Multimodal interaction summary:

- Speech used for greeting and suggesting
- Tablet used for selection and confirmation
- Pepper initiates guidance with appropriate body orientation and social distance

Non-Nominal Situation – Rejection and Preference Shift
The robot begins by asking the user whether they enjoyed the last movie they watched. The user responds negatively. Based on this feedback, the robot logs the response as negative and immediately adapts its behavior by querying the knowledge graph to generate new recommendations, exploring alternative genres. The user engages but chooses to postpone booking. In this case the robot asks whether the user would like additional assistance, for example purchasing snacks at the bar or receiving directions to cinema locations.

5.2 User Study Design and Hypotheses

To evaluate the effectiveness of our graph-based recommendation strategy, we designed a controlled user study simulating with ChatGPT 4o a typical interaction at the cinema. The aim is to assess whether the robot's personalized suggestions yield better subjective experiences than random selection.

Research Question: Does selecting a recommended movie from the robot's personalized suggestion result in higher user satisfaction than selecting a random movie?

Independent Variable:

- Movie selection strategy:
  (A) Personalized recommendation (based on the user's preferences)
  (B) Random movie (from the same available session list, with no personalization)

Dependent Variable:

- User satisfaction: Measured on a 5-point Likert scale (1 = Not at all satisfied, 5 = Extremely satisfied) collected after the movie

Null Hypothesis (H_0): There is no statistically significant difference in user satisfaction between watching a randomly selected movie and watching a movie recommended by the robot.

Experimental Protocol:

- Between-subject design with 20 participants (10 in each condition)
- Each participant interacts with the robot, provides preferences, and is assigned either:
  - a movie selected by the recommender system (Condition A)
  - or a random movie from the same screening schedule (Condition B)
- After watching the movie (or being told the premise if real viewing is infeasible), users complete a satisfaction survey
- Mean satisfaction ratings are compared between groups using a two-tailed t-test

Simulated Results:

Table 1: User Satisfaction Comparison: Recommended vs. Random Movie

User ID	Condition	Movie Type	Satisfaction (1–5)
U1	A	Recommended (Action)	5
U2	A	Recommended (Comedy)	4
U3	A	Recommended (Sci-Fi)	5
U4	A	Recommended (Animation)	4
U5	A	Recommended (Adventure)	5
U6	A	Recommended (Rom-Com)	4
U7	A	Recommended (Mystery)	5
U8	A	Recommended (Fantasy)	4
U9	A	Recommended (Family)	5
U10	A	Recommended (Drama)	4
U11	B	Random (Documentary)	3
U12	B	Random (Thriller)	2
U13	B	Random (Drama)	3
U14	B	Random (Horror)	2
U15	B	Random (War)	3
U16	B	Random (Crime)	2
U17	B	Random (Musical)	3
U18	B	Random (Sport)	3
U19	B	Random (Western)	2
U20	B	Random (Indie)	3

Statistical Analysis:

- Mean satisfaction (Recommended group): $\mu_A = 4.5$
- Mean satisfaction (Random group): $\mu_B = 2.6$
- Standard deviation (Recommended): $\sigma_A = 0.53$
- Standard deviation (Random): $\sigma_B = 0.52$
- T-test (two-tailed, independent samples): $p = 1.83 \times 10^{-7}$

Interpretation: Since $p < 0.05$, we reject the null hypothesis. The result indicates that users who selected the recommended movie reported significantly higher satisfaction than those assigned a random movie. This confirms the efficacy of our recommendation system in enhancing perceived experience and aligning suggestions with user interests.

5.3 RBC Benchmark Results

5.3.1 Functionality Benchmarking Results

The functionality benchmarking aims to evaluate essential subsystems that contribute to robust Human-Robot Interaction (HRI) in the cinema assistant scenario. Each module was tested in isolation, either through simulated environments or controlled testing sessions.

Interaction Fluency (Speech + Tablet Multimodal Flow)
Test Setup: Simulated dialogues were conducted using both speech and tablet interfaces. User inputs were either vocal (ALDialog in simulation) or touch-based, while robot outputs included synthesized speech (ALDialog in simulation) and visual prompts.
Metrics:

- **Turn-Taking Latency:** Average time between user utterance/tablet tap and robot response was 1.2 seconds.
- **Interaction Completion Rate:** 85% of sessions reached their intended goal (e.g., booking, guidance) without interruption or failure.

Observations: Delays primarily occurred during context-switching between modalities (e.g., from voice to visual output like map generation), but remained within acceptable HRI thresholds.

User Satisfaction (Subjective Evaluation)
Test Setup: After each interaction session, users completed a standardized feedback form assessing various aspects of the robot's performance.
Metrics:

- **Likert Scale Ratings:** Average score of 4.3/5 across dimensions such as friendliness, helpfulness, and clarity.
- **System Usability Scale (SUS):** Mean SUS score of 76.5/100.
- **Net Promoter Score (NPS):** 45 (indicating a favorable impression and likelihood to recommend).

Observations: Users appreciated the multimodal interaction. Negative feedback was generally tied to slow tablet responsiveness or limited speech variability.

Navigation and Guidance (Static Indoor Mapping)
Test Setup: Using a predefined cinema layout in Choregraphe, Pepper was instructed to guide users to known locations such as screens and concession stands.
Metrics:

- **Path Smoothness:** Rated "moderately smooth" on a qualitative 3-point scale, with minimal stop/start behavior or unnatural turns.

- **Completion Time:** 95% of navigation tasks were completed within 30 seconds of initiation and 100% without a time limit.

Observations: Limitations arose due to Choregraphe's inability to simulate dynamic obstacles or real-time human congestion.

Recommendation Inference (Knowledge Graph + RotatE)
Test Setup: The recommendation module was tested on a filtered subset of the MovieLens 1M dataset, with queries generated based on synthetic user profiles and historical preferences.
Metrics:

- **Mean Reciprocal Rank (MRR):** 0.16
- **Hits@10:** 0.28

Observations: The relatively low scores are partly due to the small dataset, a limitation imposed by computational constraints for on-device inference. Nonetheless, the system achieved real-time performance with acceptable latency on the robot, demonstrating practical viability under resource-limited conditions.

5.3.2 Task Benchmarking Example

Scenario A single user interacts with Pepper to complete a typical cinema visit session. The robot must recommend a movie, assist with booking, guide the user to the correct screen, and optionally recall food preferences. Performance is scored across subtasks with partial achievements considered.

1. User Identification and Preference Registration

- Time taken: 1 minute 20 seconds
- User successfully enters name, age group, and genre preferences via tablet
- Pepper stores data in the local database and confirms registration

Scoring: +5 points (completed under 1:30 min but over 1 min)

2. Movie Recommendation & Ticket Booking

- Knowledge graph queried with user's past ratings and genre preferences
- Pepper recommends an upcoming movie matching user's profile
- User accept the top recommendation but takes more than 2 minutes
- Robot confirms availability and shows session times on tablet
- User selects time and screen number is shown
- Confirmation animation and feedback collected

Scoring:

- +5 points if recommendation is accepted but booking is delayed beyond 2 minutes.

Total: +5 points

3. Navigation to Target Loaction

- Robot initiates guidance sequence after user asks to be accompanied
- Path completed in 28 seconds

Scoring: +10 points (Smooth navigation to correct screen)

4. Concession Interaction (Snack Ordering)

- Time taken: 30 seconds
- User asks about snacks
- Pepper recalls user's last preferred item: "popcorn"
- User confirms and adds it to the virtual cart

Scoring: +10 points (Personalized food suggestion successfully recovered and acted on in under 1 minute)

Overall Task Score Summary

Subtask	**Points**
User Identification and Preference Registration	+5
Movie Recommendation & Ticket Booking	+5
Navigation to Target Location	+10
Concession Interaction	+10
Total	**30 / 40**

5.3.3 World Modelling Strategy and Relation to Prior Work

Support for Social Reasoning and Adaptation.
Our world modelling approach is grounded in a hybrid representation that combines a structured spatial map of the cinema environment with dynamic user and interaction data stored in a relational database. This dual-layer model enables the robot to reason both about the physical world (e.g., room locations, distances, paths) and about social context (e.g., user preferences, history of interactions). Unlike traditional approaches that often rely on static rule-based systems or dialogue transitions, our model is updated in real time based on multimodal inputs (speech, touch, navigation events), supporting adaptation to

the user's intent and current context. While prior systems in HRI often decouple physical perception from social reasoning, our integration allows the robot to, for example, modify navigation plans or recommendations if a room (e.g., bathroom) is unavailable. Moreover, by storing feedback and preferences in a persistent knowledge base, the robot can offer socially aware and historically informed responses in future interactions.

Relation to the Literature.

Our contribution builds upon prior research in user modeling and recommender systems for HRI by extending these frameworks to a physical, embodied context. In particular, we confirm the effectiveness of leveraging past user interactions and preference similarity—an approach commonly validated in domains such as e-commerce and entertainment—but we apply it within a physically grounded scenario involving navigation, multimodal dialogue, and real-time adaptation. However, we diverge from traditional HRI recommender systems by integrating real-time spatial awareness and environmental context (e.g., room occupancy) into the recommendation process. This extension enables the robot to offer not only personalized suggestions, but also contextually appropriate ones, thereby enhancing the naturalness and robustness of the interaction.

6 Conclusion

Developing a cinema assistant with Pepper has been a stimulating and formative experience. It allowed us to move beyond theoretical models and implement a fully functioning human-robot interaction system that integrates spatial reasoning, multimodal dialogue, and real-time feedback in a unified architecture. Working on this project showed us the value of modular design and highlighted how complex behavior can emerge from the coordination of relatively simple components, when they are carefully integrated. We also observed how important spatial awareness and context sensitivity are: knowing where users are, where they want to go, and how to guide them effectively made the robot feel much more intelligent and socially competent. At the same time, we realized that robustness and fault tolerance are just as crucial as intelligence: the robot must continue functioning gracefully even when problems occur. To improve the system, several directions are worth exploring. On the technical side, integrating a lightweight vision system could enable the robot to detect facial expressions or gaze direction, enriching its perception of user engagement. Additionally, adapting navigation behavior based on real-time environmental data, such as overcrowding, could make the assistant more responsive to dynamic conditions. Beyond technical concerns, we must consider ethical and societal implications. Recording and analyzing user preferences inevitably raises privacy concerns, especially if data are stored persistently. Ensuring informed consent and favoring on-device processing are important design choices. From a societal perspective, the high cost of humanoid platforms like Pepper may limit their deployment to well-funded institutions. Exploring lower-cost platforms and open-source solutions could help democratize access to robotic assistance. Psychologically, it's

important to consider how users relate to anthropomorphic robots, and how expectations should be managed to avoid dependency or misinterpretation of the robot's capabilities. In summary, the Cinema Assistant demonstrates that a socially interactive robot can support helpful and engaging interactions in a semi-public environment. With further enhancements in perception, personalization, and ethical deployment, systems like this could evolve from research prototypes to everyday companions in public spaces.

References

[1] Cynthia Breazeal. Toward sociable robots. *Robotics and Autonomous Systems*, 42(3–4):167–175, 2003.

[2] F. Maxwell Harper. The movielens datasets: History and context, 2015.

[3] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 639–648. ACM, 2020.

[4] Dirk Holz, Gian Diego Tipaldi, Thomas Rühr, Michael Bajracharya, Raj Madhavan, and Kai Arras. Evaluating capabilities of service robots. In *Performance Evaluation and Benchmarking of Intelligent Systems (PERMIS)*, pages 21–26, 2013.

[5] Luca Iocchi. Lecture 3: Social signal processing. Lecture Slides, Human-Robot Interaction Course, 2023. Accessed July 2025.

[6] Luca Iocchi. Lecture 4: Multimodal interaction. Lecture Slides, Human-Robot Interaction Course, 2023. Accessed July 2025.

[7] Michiel Joosse, Thomas Sardar, Maaike A. Evers, and Ben Kröse. Robot etiquette: How to approach a person. In *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 317–324, 2013.

[8] Thomas Kruse, Amit Kumar Pandey, Rachid Alami, and Andreas Kirsch. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems*, 61(12):1726–1743, 2013.

[9] Seminar. seminars on knowledge adaptation and passive learning. seminars on knowledge adaptation and passive learning, 2023. Accessed July 2025.

[10] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations (ICLR)*, 2019.

[11] Peter Trautman, Joey Maxwell, Andreas Spalanzani, and Roland Siegwart. Robot navigation in dense human crowds: The case for cooperation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2153–2160, 2015.

[12] Astrid Weiss, Manfred Tscheligi, and Johannes Bernhaupt. A methodological variation for acceptance evaluation of human-robot interaction in public places. In *Proceedings of the 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 268–269, 2009.

[13] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 974–983. ACM, 2018.