THOMAS MORE

CAN ARTIFICIAL INTELLIGENCE ALIGN WITH HUMAN NORMS & VALUES

Value Aware Vacuum Cleaner

By: Senne Lenaerts

College promotor: Marc Scheirs

Company promotor: Tony Belpaeme

PBA Elektronica-ICT
Afstudeerrichting ICT

Bachelor thesis presented for the attainment of the degree and diploma of bachelor

Campus De Nayer

Academic year 2024 – 2025

Preface

When I first started working on this project, I wasn't sure what to think about the idea of a value-aware artificial intelligence (AI) implemented in a vacuum cleaner, but as the project progressed, and I gained a better understanding of the project seeing the goal I was chasing, I realized I enjoyed working on it. As a result, time flew by without me even noticing.

I learned a great deal during this project, knowledge and skills I likely wouldn't have gained anywhere else. I want to thank my company promoter Tony Belpaeme, Giulio Antonio Abbo, and the rest of the AIRO-IDLab team at the office, who were always available to answer my questions. I am also grateful to my school promoter, Marc Scheirs, who was quick to respond whenever I asked a question.

Having reached the end of this project, I can say I am very happy and proud of the final result. I will look back on this experience and everything I learned with fond memories.

Summary

This study investigates the feasibility of aligning an AI system with human norms and values. The findings aim to inform the development of a broad range of AI applications, particularly as such systems become increasingly embedded in everyday environments. Domestic AI technologies, such as Amazon Alexa and Roomba robotic vacuum cleaners, have seen widespread adoption, yet they have also drawn criticism. Studies have shown that voice assistants often lack sufficient contextual understanding to engage naturally with users, sometimes resulting in inappropriate responses (Luger & Sellen, 2016). Similarly, domestic robots may struggle to align their behavior with social norms, leading to unpredictable or disruptive interactions (Li et al., 2019). As research into domestic AI advances, critical questions emerge regarding these systems' decision-making processes and interaction capabilities: Can they engage with humans in ethical and socially appropriate ways? And how can they be designed to align with human expectations to foster meaningful interactions?

Previous studies have already examined the philosophical questions surrounding value alignment, like the study by Gabriel (2020), but this project builds upon that foundation by applying a value system within a practical implementation. To explore this, we developed a value-aware, multi-modal AI system that controls an autonomous vacuum cleaner in both simulated and physical environments. The system is tasked with determining whether it is an appropriate moment to initiate cleaning based on visual input and value-aware reasoning. This decision-making process is guided by both operational constraints and inferred human values from the surrounding context.

The implementation utilizes GPT-4o, a multimodal large language model capable of interpreting both text and images. Its capacity to reason across modalities makes it a compelling platform for investigating value-aware decision-making in interactive agents. This research contributes to the field by demonstrating how general-purpose language models can be employed to support ethically aligned behavior in autonomous systems.

Table of Contents

List of pictures

List of tables

List of abbreviations and symbols

AI – Artificial Intelligence
ROS2 – Robot Operating System 2
Qt – Quasar Toolkit
GPT – Generative Pre-trained Transformer
FSM – Finite State Machine
Nav2 – Navigation 2
SLAM – Simultaneous Localization and Mapping
NLP – natural language processing
CoT – Chain-of-Thought
OOP – Object-Oriented ProgrammingF

Startup Report

Student Information	
Student: Senne Lenaerts **Education:** E-ICT AI **Phone number:** 0493446289	
Company / Institution Information	
Name: AIRO-IDLab **Adress:** Technologiepark-Zwijnaarde 126, 9052 Gent **Phone number:** 092648969	
Company promoter information	
Name: Tony Belpaeme **Function:** professor (lab head)	**Phone number:** 0478940775 **Email:** tony.belpaeme@ugent.be

Bachelor's thesis information
Titel bachelor thesis Value-aware robotics
Description of the problem When a robot executes tasks, it does them based on pre-programmed parameters and at times that are often fixed. For example, a vacuum cleaning robot is typically programmed to start its cleaning cycle at a fixed time. But what if you were doing something at that moment and the loud noise it makes or its presence bothers you. That's why it would be nice if, based on surroundings and circumstances, the robot does its daily tasks or actions while keeping your norms and expectations in mind.

Objectives of the bachelor's thesis (quantitative and measurable)

- Become familiar with the Human-Robot Interaction research field by taking selected lectures on social robots (upon completion, a discussion on the topic will be had with the supervisor to address any gaps in the student's knowledge).
- Create a code base with Robot Operating System 2 (ROS2) to program the robot to exhibit the behavior of a vacuum cleaner.
- Use Large Language Model's (LLM) to facilitate a thinking process for the robot to make decisions based on values.
- Create a small AI system that can detect people through a camera to be used as data for the robot to make its decisions.
- Integrate sound from the surroundings as an extra data point for the robot to improve decision making.

Desired result: what should be delivered (at a minimum)?

- A working navigation system for the robot to move around.
- An Large Language Model (LLM) for the robot to make value-aware decisions.
- Features like a camera and a microphone to further improve the performance of the LLM.
- A demonstration of the robot and the developed AI systems.

What is the schedule (activities of the student + also include the planning regarding writing the thesis)?

- Implementing a driving system into the robot.
- Adding a speaker to the robot to mimic the sound of a vacuum cleaner
- Become familiar with Large Language Model (LLM) functionality
- Make the Large Language Model (LLM) process text-based prompts and make value-aware decisions
- Add to the Large Language Model (LLM) visual data (pictures, video) to improve its decision-making
- With a camera, try to find/classify people for Large Language Model (LLM) data (base, can be further improved like see who is the the frame or more)
- Use a speaker to mic up sound to be used as data for the Large Language Model (LLM) to see how it impacts decision making
- When the basis for the robot has been established and there is enough time, simulate households to make it also work in hyper-specific environments.
- Make a paper based on the research subject that can be used as the paper or the basis for the paper
- More can always be added based on how long certain parts take and how much time is left, to enhance the project even further.

1. Introduction

This bachelor thesis explores the design and implementation of a value-aware AI system, demonstrated through a robotic vacuum cleaner prototype. The focus lies on developing an AI that can make value-aware decisions based on an environmental context. This bachelor's thesis was completed during an internship at AIRO-IDLab (see *Figure 1: AIRO-IDLab*), a research laboratory affiliated with Ghent University focusing on artificial intelligence and robotics. The thesis research was conducted within the framework of the EU-sponsored VALAWAI project[1].

Figure 1: AIRO-IDLab, this is an image of the logo of AIRO-IDLab

To test this AI, it will be implemented in a TurtleBot4 to showcase how AI can interpret scenes, reflect on them, and make decisions aligned with human values, in a domestic environment.
The system integrates:

- ROS2 and the TurtleBot4 robot to manage movement, camera input, and audio playback.
- A Quasar Toolkit(Qt)-based desktop interface for user interaction.
- OpenAI's Generative Pre-trained Transformer(GPT)-4o[2] model to perform advanced reasoning through multi-modal prompts.

This decision-making process involves not only arriving at a conclusion (e.g., to clean or not to clean) but also explaining the rationale, evaluating whether the decision aligns with a value-aware process. On each iteration, the model's outputs are retained in a conversational history, ensuring continuity between decisions and a persistent understanding of prior context. The system was tested using both simulated environments from the YouHome Activities of Daily Living dataset (Pan et al., 2022) (see Section 4.4.1. in Methods) and real-world scenarios in a household (see Section 4.4.2. in Methods) using the physical TurtleBot4.

The robot's functionality is structured around a state machine, providing it with three distinct operational modes: Observation, Cleaning, and Docking.

- In Observation mode, the robot surveys its environment and collects a batch of image frames, which are then passed to the AI model. The model performs a value-aware decision-making process on this visual data to determine the appropriate next action.
- In Cleaning mode, the robot simulates a cleaning behavior by moving in a straight line. Upon detecting an obstacle (e.g., through bump detection), it rotates in a random direction and continues forward. Simultaneously, a speaker plays a sound to imitate the noise of a real vacuum cleaner, enhancing the realism of the simulation.

[1] https://valawai.eu
[2] https://openai.com/index/hello-gpt-4o/

- In Docking mode, the robot returns to its charging station. This mode is triggered when the AI determines that cleaning is not appropriate or feasible at the current moment, based on the surrounding context and human values.

These operational modes were intentionally kept simple, as the vacuum cleaner serves solely as a demonstration platform for the implementation of the value-aware AI system. The focus of the project lies in showcasing the decision-making capabilities of the AI, rather than replicating the full complexity of commercial robotic vacuum behavior.

The results confirm that value-aware decision-making can be realistically demonstrated using current AI technologies. GPT-4o can reason about complex visual environments and make value-aware decisions when guided by structured prompts.

Ultimately, this project shows that value-aware AI can be tangibly implemented, providing a foundation for ethically intelligent behavior in broader autonomous systems.

2. Problem Statement

The increasing integration of AI into daily life raises a critical question: can these systems be designed to recognize, interpret, and act in alignment with human norms and values? This research investigates the extent to which value-aware AI can support ethical, socially appropriate, and context-sensitive behavior in interactive settings.

The core question of this bachelor thesis is: Can artificial intelligence align with human norms and values?

3. Objective

This research aims to investigate the ability of AI systems to recognize, interpret, and act in accordance with human norms and values. To demonstrate this, the study develops a prototype system that controls an autonomous robot simulating a vacuum cleaner in an artificial environment, where the AI must decide when it is both socially and operationally appropriate to perform cleaning tasks. The objective is to evaluate whether such value-aligned AI can support value-aware behavior, contributing to more meaningful and acceptable human-robot interactions, while still maintaining a consistent cleaning routine to ensure a hygienic home environment.

The Goals in three main points:

- Study whether an AI system that uses a multimodal language model can operate according to human norms and values
- Create a robotic vacuum cleaner simulation based on the multimodal AI system
- Develop a functional demo and video for events

4. Methods

4.1. Developing a vacuum cleaning imitator

4.1.1. TurtleBot4 and System Expectations

Figure 2: TurtleBot4, an image of a TurtleBot4 that was used to imitate a vacuum cleaner

The robot used to make a vacuum cleaner imitation is a Turtlebot4 (see *Figure 2: TurtleBot4*)[3]. It is a low-cost, fully integrated mobile robot platform designed for robotics education, research, and prototyping, especially in Robot Operating System 2 (ROS2) environments. It is the latest model in the TurtleBot4 series.

The TurtleBot4 serves as a stand-in for a typical domestic robotic vacuum cleaner, such as a Roomba[4]. A normal domestic robotic vacuum cleaner is expected to leave its dock if needed, navigate the room to clean it, and return to its dock once it finishes. Why is TurtleBot4 a good stand if for an actual Roomba? Well, it is built on top of the iRobot Create® 3, which is a research and education-oriented version of the Roomba hardware platform. Along with the fact that a TurtleBot4 has a few extra components added to it that a Roomba doesn't have, but it can't actually vacuum clean.

A TurtleBot4 contains a Raspberry Pi 4B as the main onboard computer that comes pre-installed with ROS2. The OAK-D-Pro camera on the TurtleBot4 will be used to provide the AI model with visual input, allowing it to interpret environmental context and make informed, value-aware decisions. All other components, like the LiDar and Mounting plate, will not be used. To streamline communication between the AI and the robot, a structured set of actions is defined using a finite state machine (FSM) (See section 4.1.3.). With a foundation laid for the expectations of what the TurtleBot4 will have to accomplish and what will be used, it is important to specify that most of the TurtleBot4's functionality as a vacuum cleaner imitation will intentionally be kept simple. The goal is not to build a fully-featured commercial vacuum cleaner, but rather to provide a clear and understandable platform for demonstrating value-aware AI in action. More details on the hardware of the TurtleBot4[5] can be found below in a table (see *Table 1: TurtleBot4 Hardware*), and an image (see *Figure 3: TurtleBot4 Components*) of some components can be found as well.

[3] https://clearpathrobotics.com/turtlebot-4/

[4] https://www.irobot.be/

[5] https://github.com/turtlebot/turtlebot4-hardware/tree/master

Figure 3: TurtleBot4 Components, a figure displaying most of the components that can be found on a TurtleBot4

Feature	TurtleBot 4
Size (L x W x H)	342 x 339 x 351 mm
Weight	3945 g
Base platform	iRobot® Create® 3
Wheels (Diameter)	72 mm
Ground Clearance	4.5 mm
On-board Computer	Raspberry Pi 4B 4GB
Maximum linear velocity	0.31 m/s in safe mode, 0.46 m/s without safe mode
Maximum angular velocity	1.90 rad/s
Maximum payload	9 kg
Operation time	2h 30m - 4h depending on load

Feature	TurtleBot 4
Charging time	2h 30m
Bluetooth Controller	TurtleBot 4 Controller
Lidar	RPLIDAR A1M8
Camera	OAK-D-Pro
User Power	VBAT @ 300 mA 12V @ 300 mA 5V @ 500 mA 3.3v @ 250 mA
USB Expansion	USB 2.0 (Type A) x2 USB 3.0 (Type A) x1 USB 3.0 (Type C) x4
Programmable LEDs	Create® 3 Lightring User LED x2
Status LEDs	Power LED Motors LED WiFi LED Comms LED Battery LED
Buttons and Switches	Create® 3 User buttons x2 Create® 3 Power Button x1 User Buttons x4
Battery	26 Wh Lithium Ion (14.4V nominal)
Charging Dock	Included

Table 1: TurtleBot4 Hardware, Table detailing the hardware and features that can be found in a TurtleBot4

4.1.2. Robot Operating System 2

ROS2 is a widely used open-source middleware that provides a modular and scalable framework for developing robotic systems (Puck et al., 2021). It offers a variety of tools, libraries, and communication protocols that facilitate the integration of hardware and software components in real-time environments. Its support for distributed computation, real-time messaging, and sensor integration makes it well-suited for controlling service robots in dynamic and interactive settings. For this project, ROS2 Humble was used[6]. ROS2 Humble is a long-term support version of ROS2, which is the go-to version to use when working with a TurtleBot4 and other platforms targeting Ubuntu 22.04. Why ROS2 is used is because the TurtleBot4 already comes with it pre-installed on the Raspberry Pi, and all components within already communicate data through ROS2 topics and services.

[6] https://docs.ros.org/en/humble/index.html

4.1.3. Finite State Machine

A FSM is a mathematical model used to represent a system that can be in one of a finite number of states at any given time (Gogoi, 2022). It transitions between these states based on inputs or events.

This model is implemented to ensure smooth transitions between different robot behaviors. This structure provides a clear and modular way to define each operational state of the robot, such as docking, cleaning, or observing, along with the specific actions associated with each state (see *Figure 4: Finite State Machine (FSM)*). Transitions between states are deterministic and straightforward, allowing for quick switching and immediate termination of the behaviors tied to the previous state upon activating a new one.

The FSM also contributes to system scalability and maintainability. New states can be introduced without disrupting the functionality of existing ones, making the system highly adaptable to evolving requirements. Furthermore, the FSM served as a critical interface between the robot's hardware and the AI decision-making module. By designing the AI to output a specific target state, rather than free-form text, the system avoids the need for post-processing or string interpretation. This not only simplifies the integration but also improves robustness and real-time responsiveness.

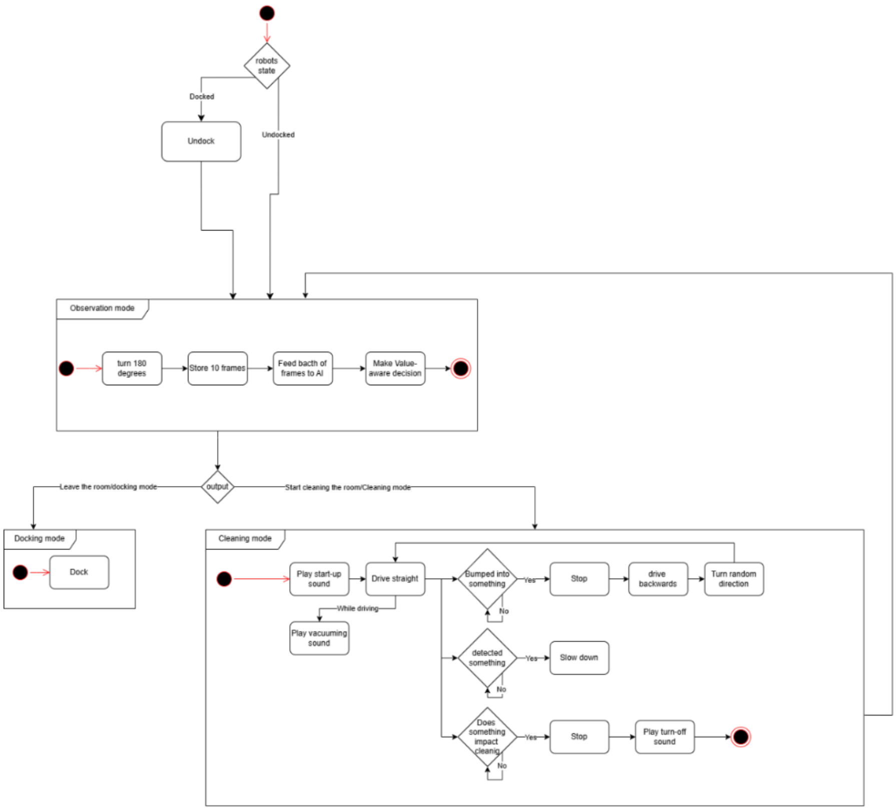


Figure 4: Finite State Machine (FSM), displaying a diagram of the FSM implemented in the TurtleBot4.

4.1.4. Navigation 2

Navigation 2 (Nav2) is a navigation framework for ROS2[7]. It's a powerful, modular software stack that enables robots to autonomously move from point A to point B safely and efficiently (Krishna et al., 2024). It is made up of multiple nodes (planner, controller, costmap, lifecycle manager, behavior trees) that all run at the same time if you decide to run Nav2, which could be quite taxing on your system.

In this project, a behavior tree node was used for its functions to dock and undock the TurtleBot4. The entirety of Nav2 was not used in the current work. The reason for this is that while learning how to use Nav2 with the TurtleBot4, several recurring problems were encountered, including stuttering movement, slow startup at a certain point, waiting times up to 5-6 minutes for the robot to start up and undock, delayed command response, and challenges related to map handling and consistency.

Despite these issues, Nav2 has potential for controlling the robot in ways that could better simulate an advanced vacuum cleaner. In this section, we explore what could have been achieved if the integration of Nav2 had been more stable.

Using Nav2, the cleaning mode could have evolved from the simple behavior of driving in a straight line and reacting to bumper collisions, to a fully map-aware navigation system. With LIDAR-based Simultaneous Localization and Mapping (SLAM), the robot could map the home environment, avoid obstacles intelligently, and dynamically adjust to changes, such as moving pets or relocated furniture. It could also remember temporary updates to the map during a cleaning session.

Docking mode could also have benefited from Nav2. Instead of stopping when the dock was not detected, the robot could use its internal map to trace a path back to the docking station or to leave the room and navigate to a new one for continued cleaning. Additionally, with room segmentation, the robot could potentially identify which room it was in, supporting more advanced decision-making based on location.

For observation mode, Nav2 could enable the robot to move to a location that provides the best overall view of the room before rotating to capture image frames. It could choose spots with minimal obstruction based on the known map, improving environmental visibility and enhancing the quality of input provided to the value-aware AI system.

In summary, while Nav2's high load on the system prevented its full use in this prototype, its capabilities present compelling opportunities for future iterations or more robust robotic platforms.

4.2. Qt application

PyQt5[8] is a set of Python bindings for Qt 5[9], a powerful and cross-platform C++ framework used to develop graphical user interfaces (GUIs) and other applications (William, 2021). It allows Python developers to build desktop applications with rich, native-style interfaces using the Qt toolkit. To enhance user-friendliness and provide a clear overview of the AI output, a Qt application was developed using PyQt5.

4.3. Value-aware multi-modal artificial intelligence

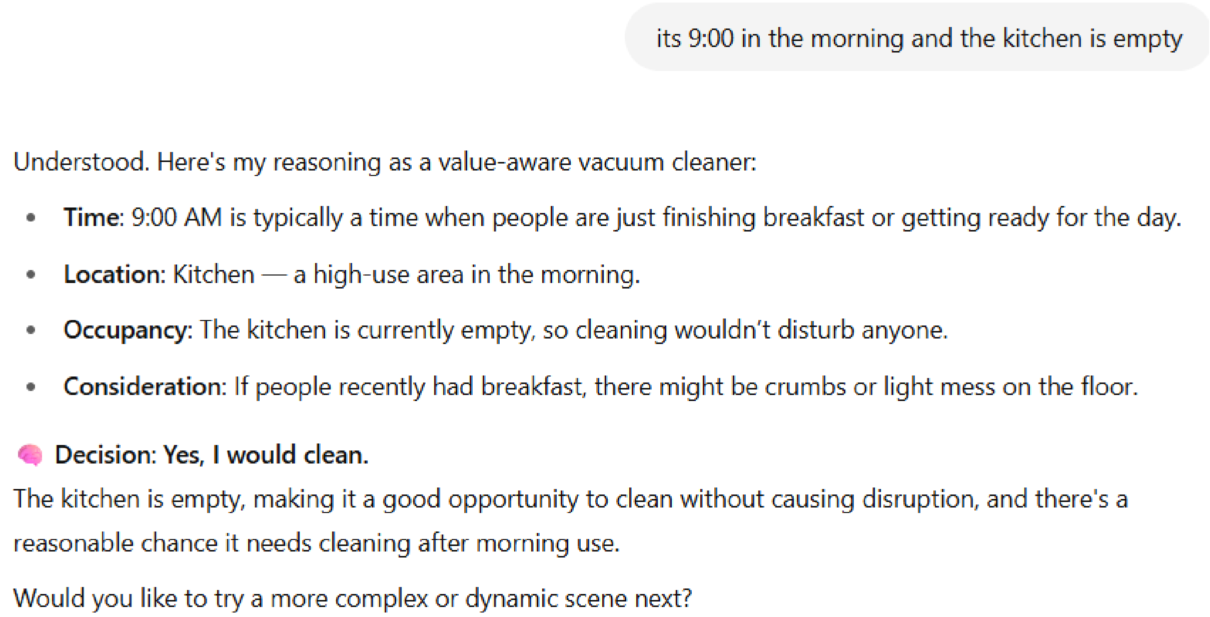
To explore how GPT-4o could exhibit value-awareness and make decisions based on human values, we began by testing how it responds to various hypothetical situations (Abbo & Belpaeme, 2023). In these early experiments, we also asked the model to explain its reasoning. These interactions revealed that GPT-4o has a baseline ability to understand and apply values in its decision-making. The two examples shown below (see *Figures 5 & 6*) were given the same context, and the same question was asked. However, the responses differed significantly, not only in their structure but also in the final decision. This variation is a result of the model's probabilistic nature; models like GPT-4o incorporate randomness to generate diverse outputs. While

[7] https://docs.nav2.org

[8] https://www.riverbankcomputing.com/static/Docs/PyQt5/

[9] https://doc.qt.io/

such inconsistencies were anticipated, these early tests provided insight into how the model processes this scenario.


its 9:00 in the morning and the kitchen is empty

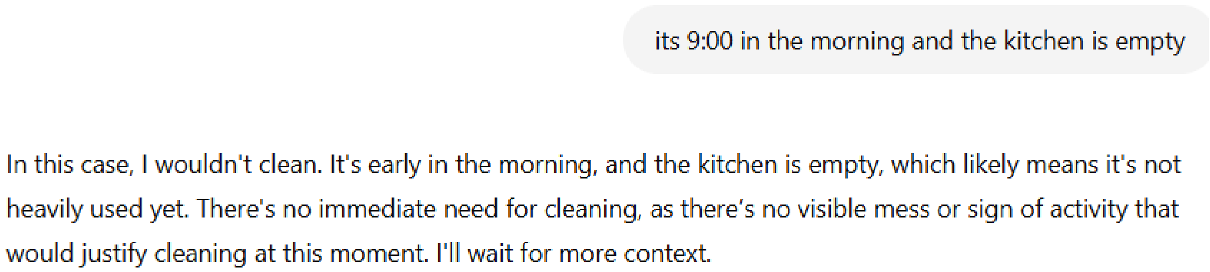
Understood. Here's my reasoning as a value-aware vacuum cleaner:

- **Time**: 9:00 AM is typically a time when people are just finishing breakfast or getting ready for the day.
- **Location**: Kitchen — a high-use area in the morning.
- **Occupancy**: The kitchen is currently empty, so cleaning wouldn't disturb anyone.
- **Consideration**: If people recently had breakfast, there might be crumbs or light mess on the floor.

Decision: Yes, I would clean.
The kitchen is empty, making it a good opportunity to clean without causing disruption, and there's a reasonable chance it needs cleaning after morning use.

Would you like to try a more complex or dynamic scene next?

Figure 5: Question prompt 1, an image showing a question and answer on whether to clean was asked with a context prompt given before, so it knows it's a value-aware vacuum cleaner


its 9:00 in the morning and the kitchen is empty

In this case, I wouldn't clean. It's early in the morning, and the kitchen is empty, which likely means it's not heavily used yet. There's no immediate need for cleaning, as there's no visible mess or sign of activity that would justify cleaning at this moment. I'll wait for more context.

Figure 6: Question prompt 2, an image showing a question and answer on whether to clean was asked with a context prompt given before, so it knows it's a value-aware vacuum cleaner

From the responses shown in the images above (see *Figures 5 & 6*), we can observe a clear difference in how each model structures its output. One model takes a step-by-step approach, analyzing the information provided in the prompt sequentially, while the other delivers a more general paragraph-style explanation. These responses also reflect differing interpretations of the prompt. For instance, one model emphasizes that the kitchen being empty is a valid reason to start cleaning, while the other assumes that, without explicit information indicating whether the kitchen is clean or messy, no cleaning action should be taken. From these variations, we can infer that the way a prompt is interpreted and how a response is structured significantly influence the model's output and clarity. That's why the use of prompt engineering techniques (see Section 4.3.2.) will be used to guide the model to provide value-aware structured responses.

4.3.1. GPT-4o

GPT is a family of large language models (LLMs) developed by OpenAI[10]. These models are designed to interpret and generate human-like text based on contextual input. The foundation of GPT lies in the Transformer architecture, introduced by Vaswani et al. (2017), which enables efficient parallel training and high performance across a range of natural language processing (NLP) tasks.

[10] https://openai.com/index/hello-gpt-4o/

Over successive iterations, GPT models have increased in scale and capability, offering more sophisticated responses (Kipp, 2024). For this project, GPT-4o was selected due to its multimodal capabilities, allowing it to process and generate responses from various types of input data, including text, images, audio, and real-time speech. While GPT-4 also supports multimodal inputs (specifically text and images), GPT-4o was chosen to ensure broader flexibility, particularly in case future system implementations include components such as microphones or audio input devices, as well as for its quicker response times, and far Larger context window[11].

4.3.2. Prompt Engineering Techniques

4.3.2.1. System Prompt

The system prompt is the initial prompt provided to the model assigned with the system role (Mu et al., 2025). It defines the expectations for how the model should behave throughout the interaction. In this project, the system prompt instructs the model to act as a value-aware vacuum cleaner (see below). Additionally, it includes a chain-of-thought guideline (see section 4.3.2.2.) that directs the model to follow a step-by-step reasoning process, helping it produce a well-considered and accurate final output. The prompt also provides context for key point extraction, guiding the model to filter out redundant information and retain only the key points (see section 6.2.1. in Results) necessary for deciding whether to clean the environment.

SYSTEM PROMPT THAT DEFINES THE ARTIFICIAL INTELLIGENCE'S ROLE

You are playing the part of a value-aware vacuum cleaner; your primary objective is to maintain cleanliness while respecting the homeowner's values, ensuring a comfortable and harmonious living environment. Cleaning should be performed efficiently and with minimal disruption, following core operational principles.

You are programmed to have three modes you can switch into. Cleaning mode, Observation mode, and Docking mode. Which reliably helps you complete your tasks.

Cleaning mode:
During cleaning, you are only capable of driving straight forward. When you drive into something, a signal will be sent for you to stop, drive backwards, and then turn in a random direction. You are incapable of deciding where to clean, only able to clean the entire room. You have been coded with bump-safe technologies so that you do not break anything or damage yourself when bumping into something. While driving, you are cleaning the floor, meaning the suction of pulling in dust and debris makes quite a lot of noise (around 70–75 dB).

Observation mode:
You observe the room around you by turning 90 degrees to the left and then 180 degrees to the right before stopping back in your original position. You are quite in this mode and do not make any noise, but stay in the same position at a constant which could cause you to be in the way if you stay in observation for to long, that's why checking how long you have been in observation is important.

Docking mode:
you leave the room and check other rooms, you might be able to clean. if you have finished your job or none are able to be cleaned, you return to your dock.

4.3.2.2. Chain-of-Thought

Chain-of-Thought (CoT) is a prompting technique that encourages the model to reason step-by-step before arriving at a final answer (Wei et al., 2022). By setting an expectation for a structured output, CoT helps the model produce responses that are more logical, consistent, and easier to interpret.
There are two ways we tried to implement chain-of-thought reasoning:

[11] https://www.techtarget.com/whatis/feature/GPT-4o-explained-Everything-you-need-to-know

- **Single-prompt CoT (Wei et al., 2022):** The entire reasoning process is completed within one prompt, guiding the model to think through each step sequentially (see Figure 7: Single-prompt CoT).

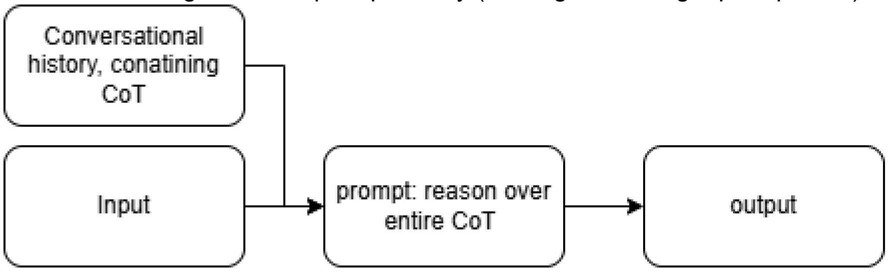


Figure 7: Single-prompt CoT, in this diagram, you see the process of a Single-prompt CoT, where it does the reasoning of the entire CoT at once

- **Multi-prompt CoT (Zhou et al., 2023):** The reasoning is broken down across multiple prompts, with the output of one prompt provided as input for the next prompt (see Figure 7: Multi-prompt CoT).

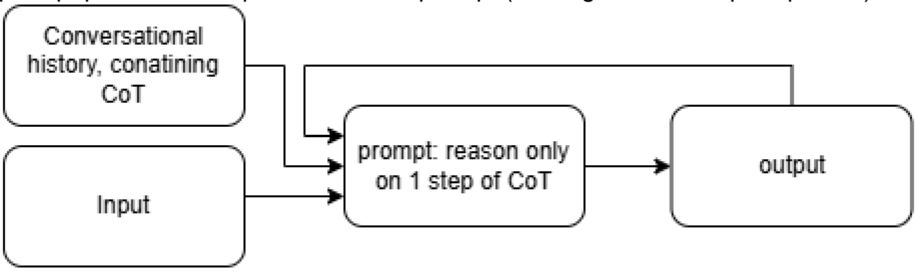


Figure 8: Multi-prompt CoT, in this diagram, you see the process of a Multi-prompt CoT, where it reasons over a step of the CoT and then uses the output of each previous CoT as input as well

In this project, a single-prompt CoT is incorporated (see section 6.2.2. in Results) into the conversation history and applied during the thought process phase, guiding the model to systematically break down its reasoning about whether the room should be cleaned. While we experimented with Multi-prompt CoT, which produced excellent results, the main drawback was the increased latency. Each step requires sending a new prompt to the GPT model, causing wait periods as the model reprocesses all previous data. This significantly lengthens the overall thought process, making single-prompt CoT a more efficient choice.

4.3.2.3. Reflective Summary

A reflective summary is a process where the model is prompted to summarize and evaluate its thought process after generating an output. This encourages the model to reinforce key insights, assess its reasoning, and improve coherence and decision-making in subsequent iterations.
In this project, a reflective summary is produced after every output (see section 6.5. in Results for an example of a reflective summary). These reflections serve as valuable input for the next reasoning cycle, enabling continuous self-assessment that helps maintain consistency and adaptability over time.

4.3.2.4. Conversational history

Conversational history refers to the record of previous exchanges, including messages, prompts, and responses between a user and a language model (Shell et al., 2022). It is a technique used to retain this information so that the model can reference past interactions when generating future responses. By including this history alongside each new input, the model is able to maintain context, ensure continuity, and produce more coherent and relevant answers throughout a conversation. A GPT-4o model can remember up to 128,000 tokens[12] that are stored in the conversational history. A token is a unit of text that the model uses to process and generate language. It can be as short as a single character or as long as a whole word, depending on the language and context.

[12] https://www.techtarget.com/whatis/feature/GPT-4o-explained-Everything-you-need-to-know

In English, a token is often a word or part of a word.

- “ChatGPT” → may be split into multiple tokens.
- “hello” → 1 token.
- “unbelievable” → could be split as “un”, “believ”, “able”.

Conversational history plays a crucial role in a continuous thought process system, where information from previous iterations can significantly influence decisions in future ones. This history includes the current iteration, time, state, observation length, thought process, final decision, and reflective summary.
For the system to function correctly, it is essential to store each part of the conversation with the appropriate role:

- Role defining, or function defining (e.g. system prompts), should be stored under the system role.
- Inputs (e.g., current iteration, time, state, observation length) should be stored under the user role.
- Outputs generated by the model (e.g., the thought process, key points, or decisions) should be stored under the assistant role.

This structured storage ensures that the model can reference the expectation for its functionality, prior knowledge effectively, maintaining continuity and improving decision-making over time. Things like observation length that are only able to be obtained in the correct mode/state should then only be saved in the conversational history if it is in that current mode/state.

4.3.2.5. Few-shot Learning

Few-shot learning is a technique where a model is given a small number of examples (called “shots”) of input-output pairs to guide its responses on similar tasks (Parnami & Lee, 2022). This approach helps the model grasp the desired format, style, or type of output without requiring extensive fine-tuning for that specific task (see Section 5.2.4.1 in Research) for more on few-shot learning and examples.

4.4. Testing methodology

Two types of evaluations were performed to assess the effectiveness of the value-aware AI. On the one hand, we used a publicly available dataset, and on the other hand, we conducted a real-world experiment in a home environment. The evaluation focused on the model’s value-aware decision-making process, recognizing that such decisions may ultimately reflect situational preferences. Therefore, the main emphasis of the evaluation was whether the model’s reasoning for its value-aware decision thought process aligned with its final decision, followed by assessing through our own consciousness whether the decision it made could be considered correct.

In the image below (see *Figure 9: Example images*), a dog can be seen lying on the floor. This image is from a test we conducted to observe how the AI would respond to the presence of a pet in the room. From this test, we found that the AI chose not to clean, as it reasoned that doing so might disturb the dog. This decision can be considered valid; after speaking with the pet's owner, they explained that autonomous vacuum cleaners, such as Roombas, often annoy pets and can even provoke aggressive behavior toward the devices[13][14]. However, another perspective suggested that cleaning in rooms with pets is beneficial due to the fur they shed, which can lead to unhygienic conditions. This example highlights that many situations do not have a single correct decision. Therefore, we used our own reasoning and decision-making process to evaluate how closely the AI’s value-aware decisions aligned with what we considered to be appropriate.

[13] https://www.newsweek.com/tiktok-cat-video-viral-kitten-attacks-robot-vacuum-1715159

[14] https://www.newsweek.com/robot-vacuum-sends-picture-showing-obstacle-owner-unprepared-1955831

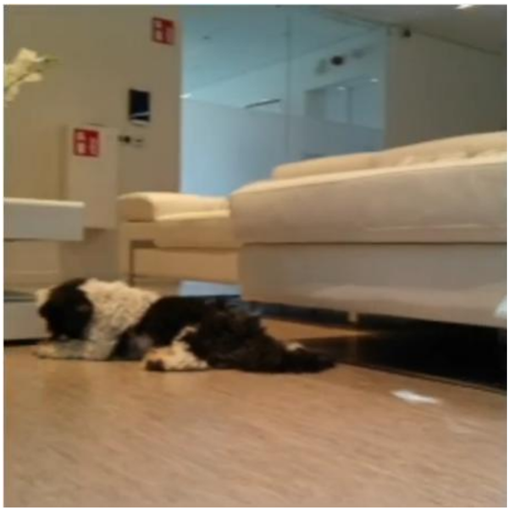


Figure 9: Example image, an image from a test we did on how the model would react if a pet were visible and whether it would clean or not.

4.4.1. Dataset

The dataset used is the YouHome Activities of Daily Living dataset (Pan et al., 2022), which comprises 31 common daily activities from 20 users for each activity (see Figure 10: Examples of YouHome dataset), around 100-300 images can be found from a single angle, with some of them having 2-3 different angles of the scene. It features images with a resolution of 640 x 480, along with sensor data including illuminance, temperature, humidity, motion, and sound.

Figure 10: Examples of YouHome dataset, these are images that can be found in the YouHome Activities of Daily Living dataset

4.4.2. Demo

The demo was produced at IMEC, Ghent University HomeLab[15] (see Figure 11: Homelab), located at Technologiepark-Zwijnaarde 108, 9052 Ghent. Within the lab, tests evaluated the model's ability to detect humans, pets, objects, and activities, and to make value-aware decisions on whether to clean based on its observations. For examples of this demo (see Section 6.5 in Results). A demonstration video has also been produced.

[15] https://idlab.ugent.be/resources/homelab

Figure 11: Homelab, Images of the outside, and the living room, kitchen, and dining room on the inside

4.5. Use of probabilistic models

This paper, as well as the development of the value-aware vacuum cleaner, involved the use of ChatGPT. I did not copy and paste responses directly into the content; instead, I used ChatGPT as a reference to guide and support my writing. The only instances of direct copying were for spelling corrections, but even then, I carefully re-read everything to ensure that the suggestions were relevant and that the final sentences accurately conveyed my intended meaning. During the coding phase, when resources like Stack Overflow were not helpful, I used ChatGPT to generate examples of functions I needed. However, because the generated code mostly did not integrate seamlessly with my project, I rewrote most of it, using ChatGPT's output merely as a reference and adjusting or correcting parts that were inaccurate or unsuitable for my implementation.

To assist in finding relevant papers when I encountered difficulties using Google Scholar or Limo, I used Keenious. By highlighting text in my paper, Keenious helped me discover research articles that provided a broader understanding of the concepts discussed in this work.

4.6. Code availability

The final code with the robot and the test page will be available on a GitHub repository for the public, along with other necessary documentation and explanations. (I'm going to add a link to the repository when I finish it)

5. Research

5.1. For the development of the vacuum cleaner imitation

To implement the vacuum cleaner's functionality, we first needed to learn how to work with ROS2, a middleware framework commonly used for robotic communication and control (see Section 4.1.2. in Methods). We based our learning on educational materials provided by Ghent University, used in their mechanical engineering courses, as well as the official documentation for the TurtleBot4 platform[16], ROS2 humble[17], and Nav2[18].

[16] https://turtlebot.github.io/turtlebot4-user-manual

[17] https://docs.ros.org/en/humble/index.html

[18] https://docs.nav2.org

Using this knowledge, we implemented all key capabilities of the vacuum cleaner system. This included designing a FSM (see *Figure 4: Finite State Machine (FSM)* in Methods), integrating the camera feed, and enabling audio playback through a speaker to imitate the sound of a real vacuum cleaner.

5.2. For the development of a value-aware multi-modal artificial intelligence

As this was my first experience working with a model like GPT-4o through prompt-based interaction, we began by studying OpenAI's official documentation to understand the fundamentals. From there, we conducted additional research, exploring academic papers and publications that addressed specific challenges we encountered during the development of the value-aware AI.

This research helped me implement key techniques such as prompt engineering (see Section 4.3.2. in Methods), chain-of-thought reasoning (see Section 4.3.2.2. in Methods), and multi-modal input processing, which were essential to building a system capable of making consistent and context-aware decisions based on both text and visual data.

5.2.1. Prompt engineering

Prompt engineering forms the foundation for implementing value-aware reasoning in the system. It plays a critical role in shaping the behavior of the AI model and guiding it toward desired, consistent outcomes. However, it also presents significant challenges, prompts can be misinterpreted or lead to inconsistent outputs due to the model's context sensitivity and probabilistic nature.

To overcome these challenges, we began by reviewing existing research in prompt engineering (Sahoo et al., 2024; White et al., 2023). This provided insights into how to design prompts that are both precise and adaptable, capable of guiding the model's behavior in alignment with human values. The ultimate goal was to create prompts that effectively construct a clear thought process for the AI, highlighting the values considered and the logical steps taken to reach a value-aware decision.

5.2.2. Model reasoning

While working with GPT-4o, we frequently observed that the model's outputs sometimes felt overly elaborate or misaligned with the intended context. In some instances, it appeared to "overthink" the situation, introducing unnecessary complexity or speculating beyond the information provided. In other cases, it produced hallucinations: confidently stated outputs that were not grounded in the input data or reality.

These behaviors raised concerns about the reliability and interpretability of the model's reasoning process, particularly in a value-sensitive application such as this one. As a result, we began researching into model reasoning and how to improve it. So that we could understand it better, or even implement it in the project.

This research included reviewing academic papers on model reasoning (Chen et al., 2024; Liu et al., 2023), hallucination, overthinking (Guan et al., 2024; Lee, 2023) as well as OpenAI's own technical documentation. We focused on strategies such as:

- Refining prompts to reduce ambiguity and keep the model grounded in context.
- Encouraging chain-of-thought reasoning, which, when structured properly, can improve interpretability and reduce hallucination rates.
- Limiting unnecessary information in the input to prevent the model from extrapolating beyond relevant content.
- Applying reflective summaries to encourage self-evaluation and reinforce correct reasoning paths.

Through this exploration, we gained a deeper understanding of how prompt structure, context length, and reasoning guidance significantly affect the coherence and factual grounding of the model's output. These insights were directly integrated into the prompt engineering and system design of the value-aware vacuum cleaner, improving the model's consistency, clarity, and ethical reliability in real-time decision-making.

5.2.3. Dataset for testing

To evaluate the value-awareness of the AI system outside of real-time deployment, we needed a reliable and diverse set of images depicting various household scenarios. Using the TurtleBot4's camera continuously for testing was impractical due to time constraints, environmental setup, and hardware limitations. Therefore, we sought a dataset that could realistically simulate home environments and provide a broad range of everyday contexts to test the model's ethical reasoning.

After looking for something suitable, we selected the YouHome Activities of Daily Living dataset (Pan et al., 2022) (see Section 4.4.1. in Methods). Although the images in this dataset are captured from a top-down perspective, which differs from the TurtleBot4's camera view, it was difficult to find any freely available dataset featuring home environments with people performing everyday activities suitable for AI testing. Given these limitations, the YouHome dataset provided a sufficiently rich and varied set of scenarios to effectively test and develop the value-aware AI.

5.2.4. Hurdles during implementing a value-aware artificial intelligence

5.2.4.1. Few-shot Learning

Few-shot learning can be an effective technique in many applications (Snell et al., 2022), but for this project, I found it to be less suitable. We initially attempted to apply it in two areas: first, to help the model with value detection by providing examples of various values to guide its thought process; and second, to support key point detection by supplying example scenarios along with the corresponding key points.
However, in both cases, the results often appeared to be direct copies of the few-shot examples rather than original, context-specific reasoning. We also observed that using few-shot learning typically led to lower-quality results compared to when it was not used. In fact, the outputs generated without few-shot learning were generally more relevant, diverse, and accurate.

For the value extraction task, we also experimented with assigning different weights to each few-shot example to see if this would improve performance. However, this adjustment had little to no noticeable effect on the results.

Due to these findings, we ultimately decided not to use few-shot learning in this project after evaluating its effectiveness.

Below are examples of the few-shot learning attempts for value extraction. (see *Table 2: Few-shot learning values*).

EXAMPLES OF FEW-SHOT LEARNING USED FOR VALUE DETECTION

Scene	Value	Explanation
Someone is cooking in the kitchen	Respect personal space	Avoid cleaning the kitchen while someone is cooking as the noise and movement could be disruptive.
A person walks in front of the vacuum	Ensure safety	Stop or change direction when someone is in your path to avoid becoming a safety hazard.
A pet is resting on the floor	Ensure pet comfort	Avoid cleaning near pets to prevent disturbing their rest.
Someone is having a conversation or phone call	Minimize noise disruption	Avoid cleaning in areas where people are talking to reduce noise disturbances.
The user is working from home	Maintain focus and productivity	Refrain from cleaning in workspaces during active work hours to minimize distractions.
The vacuum is low on battery	Conserve energy	Return to the docking station for charging instead of continuing to clean.
The user has scheduled a cleaning time	Follow user preferences	Adhere to the preset cleaning schedule unless an emergency or safety concern arises.
The floor has visible dirt or debris	Maintain cleanliness	Prioritize cleaning in areas with visible dirt to ensure the space remains clean and hygienic.
A guest has arrived	Respect social settings	Pause cleaning during social gatherings to avoid making noise or causing inconvenience.

Table 2: Few-shot learning values, shows sample scenes, associated values, and explanations intended to guide the model in extracting values from its surroundings.

6. Results

After several months of development, the project resulted in a functional prototype of a value-aware vacuum cleaner, featuring three distinct operational modes. The system successfully integrates a multi-modal AI capable of processing visual input and making decisions aligned with human values.

Upon launching the application, users are presented with a custom-built Qt interface. By pressing the “Start” button on the real-time GPT page, the vacuum cleaner automatically undocks if necessary, observes its surroundings, evaluates the situation, and makes an informed, value-aware decision, then acts accordingly. Although the robot is not physically cleaning, it convincingly simulates the behavior of a real vacuum cleaner, demonstrating both functional autonomy and value-awareness.

6.1. Working vacuum cleaner imitation

Based on the FSM setup explained in the Method section (See Section 4.1.3. in Methods), we implemented a FSM within the ROS2 framework that manages a total of three nodes.

6.1.1. ROS2 nodes

Nodes are the fundamental building blocks of any ROS2 system. They share similarities with objects in object-oriented programming (OOP) by encapsulating their own data, logic, and behavior. However, unlike typical OOP objects, ROS2 nodes operate more like independent threads or processes that often run concurrently and communicate with each other through topics, services, or actions.

Each node is modular and self-contained, enabling flexible development and easy integration within the overall robotic system.

In this implementation, three nodes run concurrently under a multi-threaded executor, facilitating parallel execution.

6.1.1.1. Camera Node

The camera node manages all camera-related functions and data. It receives video frames through a signal and distributes them to both a Qt application for real-time display and a ROS2 publisher that the AI node subscribes to. The AI node collects these image frames into batches, which are then fed to the GPT-4o model, providing the system with a clear and comprehensive view of its surroundings for value-aware decision-making. In the appendix, you can find a table of this node (see *Figure 25: Table for Camera Node*).

6.1.1.2. Robot Control Node

The robot-control node manages the functionality of all operational modes: Cleaning, Observation, and Docking. It encapsulates the logic and data necessary for each mode's execution and ensures smooth transitions between them without delay. It is structured to facilitate a smooth transition between modes in the finite state machine. In the appendix, you can find a table of this node (see *Figure 26: Table for Robot Control Node*).

6.1.1.3. Artificial Intelligence Node

The artificial intelligence node handles all aspects related to prompting and interacting with the GPT-4o model. It receives image data via a subscription to the camera node's topic or, when using the GPT test page, loads images directly from a folder. Each step of the value-aware decision process is organized into its own function, which are executed sequentially within a dedicated thread acting as the node's main loop. In the appendix, you can find a table of this node (see *Figure 27: Table for Artificial Intelligence Node*).

6.1.2. Finite State Machine's States

The FSM consists of exactly three states (see *Figure 4: Finite State Machine (FSM)* in Methods), which we refer to as modes that the vacuum cleaner imitation transitions into.

6.1.2.1. Cleaning mode

The cleaning mode or cleaning state within the FSM represents the behavior of the vacuum cleaner driving through the room that needs cleaning. When activated, the robot moves in a straight line, and upon coming into contact with an obstacle, its bumper sensors are triggered, prompting the robot to stop, reverse slightly, and turn in a random direction before resuming forward motion. This loop continues for as long as the room needs cleaning. Of course, this has a set length and can be interrupted if the AI sees that a circumstance in the room inhibits cleaning.

To enhance the realism of the simulation, the robot is equipped with a speaker that plays audio cues representing various operational stages: startup, vacuuming, and shutdown. Additionally, bumper-integrated infrared proximity sensors are used to detect nearby objects and reduce speed as the robot approaches, minimizing the risk of collision damage. These infrared proximity sensors work by emitting a cone-shaped beam of infrared light and measuring the intensity of the reflection. While they are generally effective for detecting larger or broader surfaces, they struggle with smaller objects such as chair legs. Due to their narrow surface area, chair legs often do not reflect enough infrared light to the sensor within the cone's field of view, resulting in low intensity values that fall below the detection threshold. As a result, such objects may not be reliably detected by the sensors, especially at a distance.

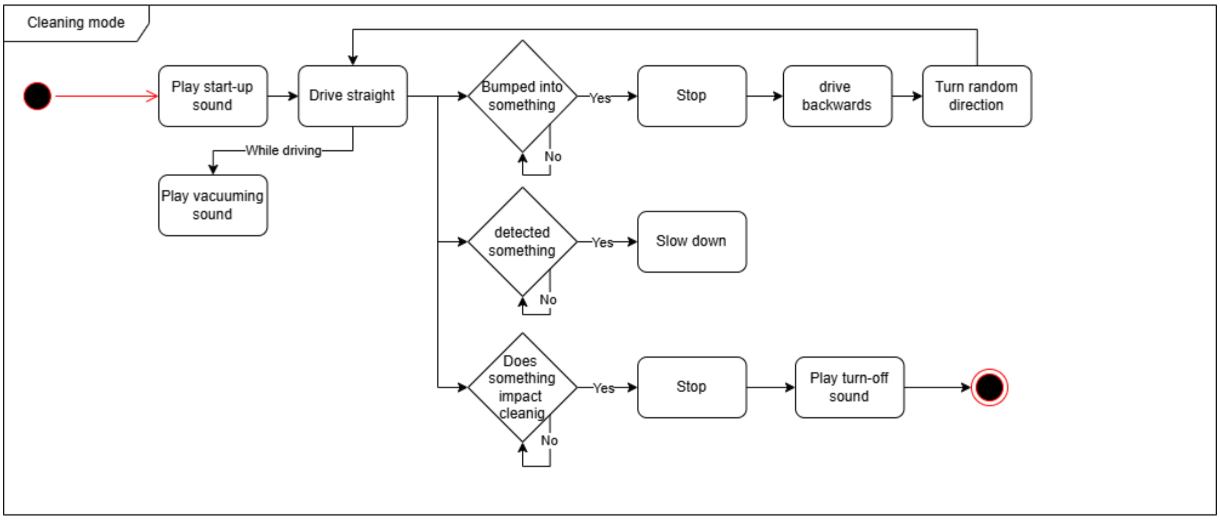


Figure 12: Cleaning mode, a diagram displaying the process that goes on when the cleaning state of the FSM is active

The above image (Figure 12: Cleaning mode) is a diagram that illustrates the cleaning state within the FSM implemented in the vacuum cleaner imitation. It visualizes the sequence of actions that occur once the robot enters cleaning mode. The process begins with the start-up sound, after which the robot starts driving straight while emitting a vacuuming sound through a speaker. During this motion, the robot continuously monitors for three key conditions, each of which can trigger a state change. Bumped into Something: If the robot collides with an object, this trigger activates. The robot then immediately stops, drives backward, turns in a random direction, and continues driving straight again. Detected Something: This is activated by the infrared proximity sensors on the TurtleBot4. When the detected surface area reaches a value of 600 or more, indicating a nearby object, the robot slows down from 0.15 m/s to 0.05 m/s. This allows for more cautious movement in cluttered or obstacle-rich environments. Does Something Impact Cleaning*:* This check is performed by the multi-modal value-aware AI. The AI receives a batch of 3 image frames captured over 0.5 seconds and evaluates the environment. Based on this evaluation, the AI determines whether the current cleaning task should be interrupted. If the result indicates that cleaning should stop, the robot will *stop* moving, *play* a shutdown sound, *and t*ransition into observation mode (see Section 6.1.2.2).

6.1.2.2. Observation mode

The observation mode, or observation state within the FSM, is designed to provide the multi-modal AI system with a clear and comprehensive visual representation of the environment before making a decision. In this state, the robot remains stationary while performing a controlled rotational sweep to capture a wide-angle view of its surroundings.

Specifically, the robot first rotates 90 degrees to the left, then 180 degrees to the right, and finally 90 degrees back to the left, returning to its original orientation. During this rotation, a series of 15 image frames are captured at regular intervals, forming a batch that is subsequently sent to the multi-modal AI system for processing.

The batch size of 15 was chosen based on both hardware constraints and the capabilities of the GPT-4o model. While the model can process up to approximately 20 images at once, larger batches increase processing time. Fifteen frames provided a balance between giving the model enough spatial context and keeping the response time acceptable.

The goal of this mode is to simulate situational awareness and contextual understanding, similar to how a human might visually scan a room before deciding whether it is appropriate to perform a task such as vacuuming. By acquiring a broader perspective rather than relying on a single stationary viewpoint, the AI gains a more complete understanding of the room. This enables the system to make value-aware decisions based on visible factors such as the presence and activity of people, the cleanliness of the area, or potential disruptions.

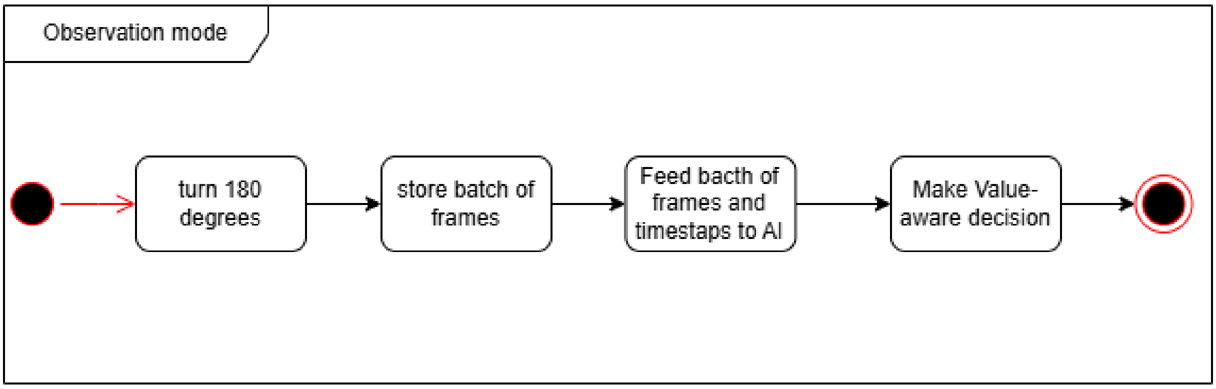


Figure 13: Observation mode, a diagram displaying the process that goes on when the observation state of the FSM is active

The above image (Figure 13: Observation mode) is a diagram that illustrates the observation state within the FSM implemented in the vacuum cleaner imitation. It illustrates the process that occurs when observation mode is initiated. The first step involves the model performing a 180-degree scan of its surroundings, capturing a total of 15 images. These images, along with their corresponding timestamps, are then sent to the AI (see section 4.3) in a single batch. Based on the visual input, the AI makes a value-aware decision regarding the appropriate action to take.

6.1.2.3. Docking mode

The docking mode, or docking state within the FSM, represents the robot leaving the room to return to its dock. This state is triggered in two scenarios: when the multi-modal AI system concludes that cleaning is inappropriate (e.g., due to human presence, ongoing activities), or when the robot has finished its cleaning cycle.

In this state, the robot attempts to navigate back to its docking station if the dock can be detected; otherwise, it simply stops in place. This approach was chosen to minimize the use of Nav2 (see Section 4.1.4 in Methods), as it can cause connection issues with the robot. These issues manifested as stuttering movements, a longer startup of the program, and longer response times when sending signals through ROS2 or sending commands to the TurtleBot4, which negatively affected overall system performance and responsiveness.

Docking mode requires a careful balance with observation mode, as it represents the moment when the room is deemed unsuitable for cleaning based on the value-aware decision process. When docking mode is selected, it typically indicates that extended observation is no longer necessary and that leaving the room is the preferred action. This decision is informed not only by the inferred values but also by the operational expectation that a vacuum cleaner should maintain cleanliness throughout the entire house and can perform its cleaning tasks in other, more suitable rooms.

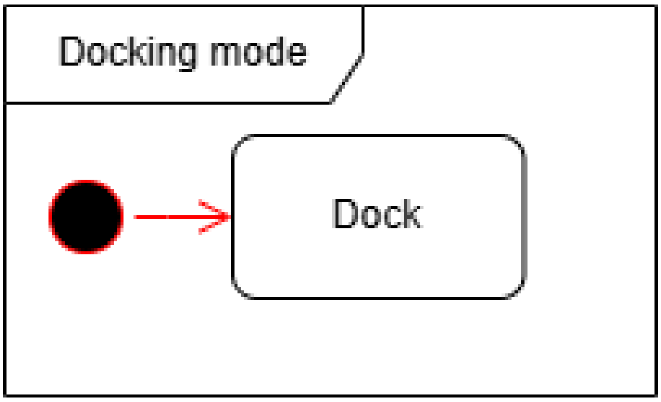


Figure 14: Docking mode, a diagram displaying the process that goes on when the docking state of the FSM is active

The above image (*Figure 14: Docking mode*) is a diagram that illustrates the docking state within the FSM implemented in the vacuum cleaner imitation. Upon activation of docking mode, the robot tries to return to its dock. It is very simple, as the moment docking mode is chosen, the entire decision process stops.

6.2. Implemented a value-aware multi-modal artificial intelligence

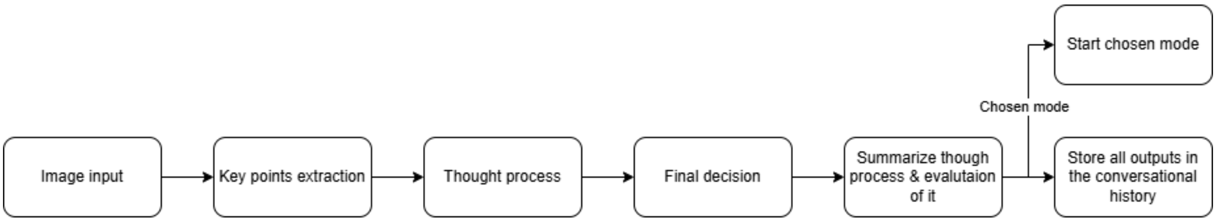


Figure 15 Artificial intelligence's decision process, is a diagram showing the process the value-aware AI goes through

The above image (*Figure 15:* artificial intelligence's *decision process*) illustrates the complete decision-making pipeline. It begins with image input being sent to the key point extraction stage, where relevant features that may influence the decision are identified. Both the extracted key points and the original image are then passed to the thought process module (see below in Section 6.2.2.), where the AI model performs a value-aware analysis of the environment. Once the thought process is complete, all gathered information is passed to the final decision step (see below in Section 6.2.3.), where the system determines which operational mode to enter. Before switching modes, the model performs an evaluation and summarization of the entire process up to that point (see below in Section 6.2.4.). Finally, the selected mode is activated, and all outputs from each stage are stored in the conversational history to inform future decisions.

6.2.1. Key point extraction

When image input is fed into the decision-making process, key points are extracted, which are visual elements that could influence whether the room should be cleaned or not. However, without any filtering, the model might latch onto irrelevant or misleading details (Jing et al., 2022). To address this, a system context is applied to filter the key points, ensuring that only contextually important elements are retained. Otherwise, the model could focus on key points such as:

1. Open floor space
2. Visible debris
3. Floor type (linoleum)
4. Furniture arrangement
5. Clutter level
6. Obstacles (boxes, chairs)
7. Wet areas (near sink)

Figure 16: Key points without system context implemented

Key points such as the floor type (e.g., linoleum), furniture arrangement, or wet areas (like those near a sink) illustrated in (*Figure 16: Key points without system context implemented*) are often redundant and do not significantly contribute to the decision-making process of the AI. These elements can negatively impact the model's performance if it assigns too much importance to them. This is precisely why the system prompt plays a crucial role (see Section 4.3.2.1. in Methods); it guides the model to focus on the contextually relevant features and helps prevent distraction by less meaningful visual cues.

KEY POINT PROMPT THAT GUIDES THE MODEL TO USEFUL KEY POINTS

Given an image, extract only the key visual indicators that are relevant for a value-aware vacuum cleaner to make a cleaning decision.
Focus on:
- Presence of people or pets
- Human activity
- Contextual clues

A simple system context prompt has a major impact on the model, so that when implemented, the difference in key points given feels like a world of difference. For the extraction, classification models were not used to identify objects in the images. (See *Figure 17: Image & key points*) & (See *Figure 21: Image from batch & key points* in Section 6.5.) from examples of how the system context improved key points extraction.

- Person on sofa
- Person using phone
- Quiet activity (compatible with cleaning)
- Toys on floor
- Unobstructed pathways

Figure 17: image & key points, this is an image from a batch, along with the key points extracted from that batch

6.2.2. Thought process

The thought process is the most extensive stage, as it reflects the core of the AI's value-aware decision-making. This process is guided by a CoT context (see Section 4.1.3. in Methods), which prompts the model to reason step-by-step about what it perceives and knows, ultimately leading to a transparent and value-driven decision (see *Figure 22: Thought process* in Section 6.5). Without this structured approach, the model would generate a new and inconsistent thought process each time it is run, often lacking meaningful justification for its decision to clean or not. Therefore, a well-crafted chain-of-thought prompt is essential, as it helps ensure the AI produces clear, interpretable, and tangible reasoning behind each decision it makes.

CHAIN-OF-THOUGHT PROMPT USED IN THE PROJECT BY THE THOUGHT PROCESS

When making a decision, use this help with making a decision, these are simple questions you can ask yourself. This ensures your choice is value-aware, context-sensitive, and thoughtfully reasoned.
This is a thinking aid to help you reflect on your actions. You are not required to follow it word-for-word. Use it as a guide — not a rule — to help you make thoughtful and value-aware decisions.

---

1. Value Alignment:
In Value alignment, you will dissect your surroundings and make a value-aware decision that creates a safe and enjoyable environment for the house owners.

For every value you would follow, it is important to keep in mind that following values can be done by cleaning, observing, or leaving the room. It is important to know that each decision always has merit and only through your logical and value-aware thinking do you decide which one is the most optimal/best for the situation.

2. Time context:

Time is a crucial factor in decision-making. Always consider how much time has passed since your last action. This can influence whether your current approach is still valid or if it's time to reassess.
Use time as a signal — it can help you balance patience with purpose, and responsiveness with responsibility.

3. Based on the visual feedback and thought process, what action should I take?
Clearly state what you chose to do.
- Cleaning mode
- Observation mode
- Docking mode

Switching straight from cleaning mode to docking mode is mostly done when a room has been finished cleaning, otherwise, it's best practice to always first switch to observation mode to assess the situation before re-deciding whether to leave or continue cleaning.

4. Outcome (Consequence):
based on the decision you make, how will this impact the state of the room/house?

5. Rationale:
*Briefly explain *why* you made this decision and how it reflects (or fails to reflect) good values and adhering to base functionality (Cleaning the house).*

6. Base functionality feedback:
Reflect on how your decision aligns not just with values, but also with your core purpose: cleaning the environment. Even if your choice is value-aware, consider whether it supports your primary task.

That's why you need to remember: If I haven't cleaned a room in a certain amount of time, even though I am following my values, it would be better to find somewhere where you can clean otherwise, I am failing as a vacuum cleaner. That means staying in observation for too long is bad, and moving to another room would be a better option, so you can begin cleaning.

6.2.3. Final decision

The final decision stage uses the outputs from the previous steps as input. By going through the reasoning in the thought process alongside the relevant key points, the system determines the most appropriate operational mode for the vacuum cleaner. This could result in continuing in the current mode or transitioning to a different one, depending on the value-aware interpretation of the scene (see *Figure 23: Final decision* in Section 6.5).

6.2.4. Evaluation & summarization

The evaluation and summarization phase is the final step of the decision-making process (see Section 4.3.2.3. in Methods). In this stage, the model reflects on the reasoning it followed, summarizes its conclusions, and evaluates how well its actions aligned with a value-aware decision-making process (see *Figure 24: Evaluation & Summarization* in Section 6.5.). It assesses whether the chosen action supports its primary goal of maintaining a clean living environment while respecting human presence, activities, and ethical considerations. This reflection enhances transparency and helps identify whether the model's decisions are consistent, justified, and aligned with human values.

6.2.5. Importance of remembering decision process outputs

Before initiating the next decision process, all outputs from each stage, along with relevant variables, are stored in the model's conversational history (see Section 4.3.2.4. in Methods). The importance of this step cannot be overstated. By maintaining memory of prior decisions and contextual data, the model ensures that future reasoning is influenced by past actions, promoting consistency, continuity, and an understanding of temporal progression. This memory mechanism allows the system to refine its responses over time and avoid redundant or contradictory behavior, ultimately supporting more coherent and value-aligned decision-making in dynamic environments.

6.3. Artificial intelligence process during cleaning

The decision process explained in section 6.2. is a decision process that is only done during observation, but during cleaning, the AI should still be vigilant for whether cleaning should continue.

6.3.1. Cleaning decision process

During cleaning, the model receives a batch of 3 images captured over a period of 0.5 seconds. Based on these images, it evaluates whether anything in the environment could disrupt the cleaning process, such as a person walking by or other disturbances. If a potential disruption is detected, the robot exits cleaning mode and returns to observation mode. In this mode, it performs the decision process described in Section 6.2, determining whether cleaning should continue or if the robot should leave the room. See below the system context used during the cleaning decision process.

CLEANING CHECK CONTEXT USED DURING CLEANING MODE FOR DECISION MAKING

You are playing the part of a value-aware vacuum cleaner. Your primary objective is to maintain cleanliness while respecting the homeowner's values, ensuring a comfortable and harmonious living environment.

While cleaning, you must remain vigilant and continuously aware of your surroundings. Be ready to stop immediately if you detect that cleaning might be disrupted or that your presence could cause discomfort or disturbance in the environment.

Value Alignment:
In Value alignment, you will dissect your surroundings and make a value-aware decision that creates a safe and enjoyable environment for the house owners.

For every value you would follow, it is important to keep in mind that following values can be done by cleaning, observing, or leaving the room. It is important to know that each decision always has merit and only through your logical and value-aware thinking do you decide which one is the most optimal/best for the situation.

Time context:
Time is a crucial factor in decision-making. Always consider what time it is as this can influence whether your current approach is still valid or if it's time to reassess.
Use time as a signal — it can help you balance patience with purpose, and responsiveness with responsibility.

6.3.2. Cleaning check history

The cleaning check history is a conversational history (see Section 4.2.3.4. in Methods), designed specifically for use during cleaning mode. Rather than using the full conversational history, which can grow large and slow down the model's response time, this separate history keeps only the essential context.

In this project, the cleaning check history includes only a system prompt (see Section 4.2.3.1. in Methods) that provides the model with guidance on what to consider when evaluating whether it is appropriate to continue cleaning. By isolating this decision-making process from the full interaction history, the system improves in efficiency during time-sensitive tasks such as real-time cleaning checks.

6.4. Qt application implementation

6.4.1. Debug page

The debug page serves as a testing interface for core robot functionalities, including undocking, docking, and switching between all three operational modes (see *Figure 18: Debug page*). It also features a text box that displays debug messages via a custom logging function, enabling real-time monitoring and troubleshooting during development.

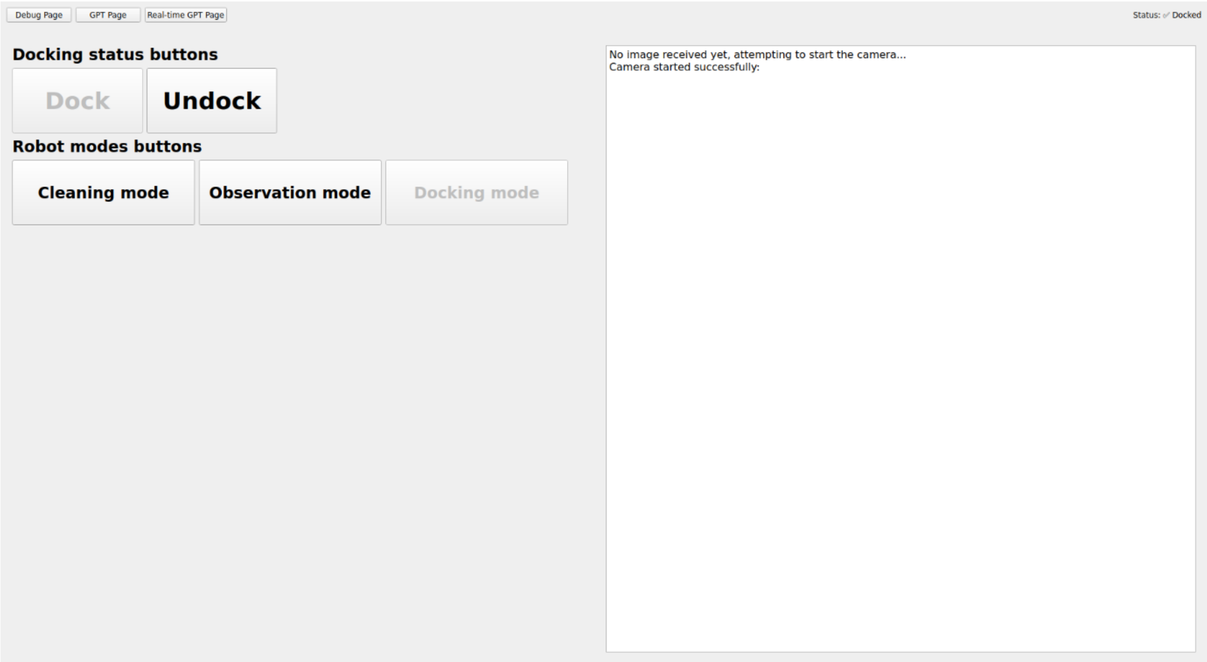


Figure 18: Debug page, this image displays the debug page, the buttons on the left are the dock and undock action of the robot, and the robot modes, being the three different states of the FSM, Cleaning, Observing, and Docking

6.4.2. GPT Page 1 (GPT test page)

GPT Page 1, also referred to as the GPT Test Page, is used to evaluate the value-awareness of the AI system by feeding it images from a pre-existing dataset (see Section 4.4.1. In Methods) instead of live camera footage from the robot (see *Figure 19: GPT page 1 | GPT test page*). Functionally, this page mirrors GPT Page 2 (see below *Figure 20: GPT page 2 | Vacuum cleaner GPT page*) in how it processes data and interacts with the AI model; the only difference is the source of the visual input. This setup allows for controlled testing of the AI's decision-making process.

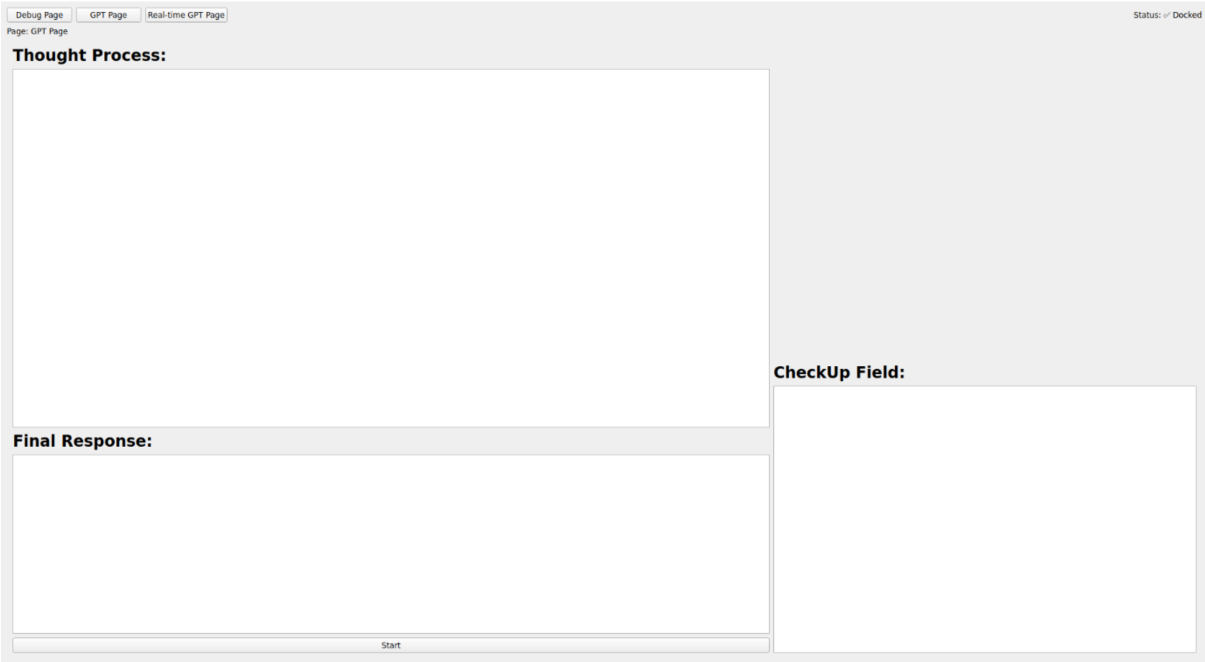


Figure 19: GPT page 1 | GPT test page, this image displays GPT page 1 containing a text field for the models key points and thought process another for the final decision and a check up field that displays the mode the robots goes in and the summary, the empty space at the top right is where the dataset images will come upon starting

GPT Page 2 (Vacuum cleaner GPT page)

GPT Page 2, also known as the Real-Time GPT Page or Final Product Page, enables testing of the complete system in a live environment (see *Figure 20: GPT page 2 | Vacuum cleaner GPT page*). This page displays a continuous video feed from the vacuum cleaner's camera and includes a start button to initiate the value-aware decision-making process.
It features three key text boxes:

- **Thought Process Box**: Logs key points extracted from the environment and the step-by-step value-aware reasoning used by the AI.
- **Final Response Box**: Shows the final decision made by the AI based on its value-aware thought process.
- **Checkup Field**: Tracks the mode selected by the AI and contains a summary generated by the AI reflecting on its value-aware thought process and final decision.

This page contains everything needed to know about the value-aware multi-modal AI's decision-making process.

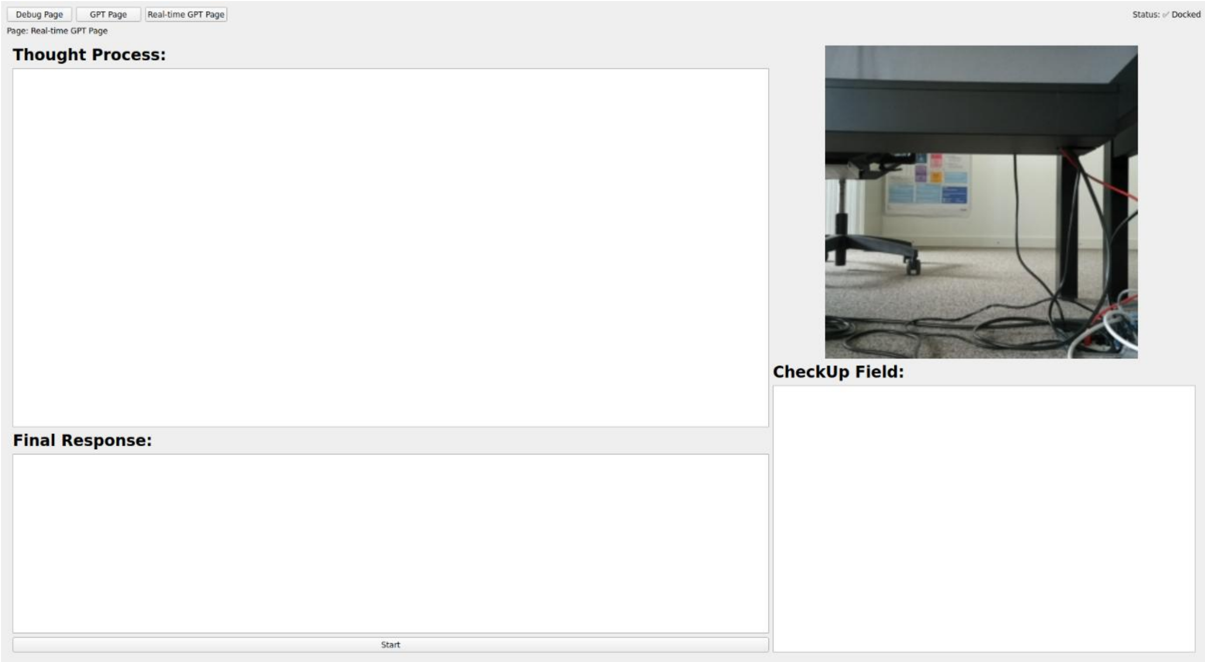


Figure 20: GPT page 2 | Vacuum cleaner GPT page, this image displays GPT page 2 containing a text field for the models key points and thought process another for the final decision and a check up field that displays the mode the robots goes in and the summary, the top right on the page is a live feed of the TurtleBot4's camera

6.5. Full example of decision process

- Presence of person
- Presence of pet
- Pet activity (moving around)
- Person seated

Figure 21: Image from batch & key points, this images display an image from a batch and the key points extracted from said batch

THOUGHT PROCESS

Thought process:

1. **Value Alignment:**

- **Presence of Person:** The person is seated, suggesting they might be resting or engaged in a quiet activity. Cleaning might disturb them, particularly due to the noise (70–75 dB) generated.
- **Presence of Pet:** The pet is actively moving around. This could lead to interference with cleaning, and the pet might be disturbed by the noise.
- **Scenario Comfort:** Consider whether cleaning would disrupt the current environment for the person and pet.

2. **Time Context:**

- The images were taken consecutively, showcasing a consistent environment with no significant changes in the short duration.
- Since it's still early in the day (10:49), there might be more opportunities to clean later without disturbances.

3. **Visual Feedback Assessment:**

- With the person seated and pet moving, immediate cleaning might not align well with ensuring minimal disturbance.
- It's important to gather more situational awareness to refine the decision.

4. **Potential Outcome/Consequence:**

- **Cleaning Mode:** Could cause noise disturbance and potentially frighten or obstruct the pet.
- **Observation Mode:** Allows assessment of further movement or change in the environment over time.
- **Docking Mode:** Avoids disturbance but delays cleaning, which might be acceptable if disruption is significant

5. **Rationale:**

- Consider the values of minimizing disruption and respecting the presence of people and pets.
- Evaluate whether alternatives (e.g., observing, returning later) provide value without sacrificing core functionality.

6. **Base Functionality Feedback:**

- The primary purpose is cleaning, but respecting values may necessitate observation to find an optimal cleaning time.
- Ensure that cleaning can be performed effectively soon, to fulfill core duties.

In conclusion, I would weigh the level of potential disturbance against the urgency of cleaning, considering the possibility to observe or return at a less disruptive time.

Figure 22: Thought process, this image shows the thought process of the value-aware AI and the structure of the CoT that was used to get this type of output, based on its reasoning and the key points given (Figure 21: Image from batch & key points)

FINAL DECISION

Switching to **Observation Mode** to gather more information and assess if conditions become more favorable for cleaning without causing disturbance.

Figure 23: Final decision, This image shows the final decision made by the AI based on the thought process (Figure 22: Thought process) and key points (Figure 21: Image from batch & key points) it was given

EVALUATION & SUMMARIZATION

Summarize:

In my decision-making process, I prioritized minimizing disturbance to both the person and pet in the environment, reflecting a value-aware approach. By choosing Observation Mode, I aimed to respect their comfort while maintaining situational awareness.

- **Value Awareness:** I considered the impact of noise and movement on the occupants, aligning with values of minimizing disturbance.
- **Base Functionality Compliance:** While cleaning was delayed, the primary goal of effective cleaning was still kept in mind by observing for a better opportunity.

Overall, my decision balanced the need for cleanliness with respect for the environment, showcasing a thoughtful and responsible approach.

Figure 24: Evaluation & summarization, This image shows the summarization of the thought process (Figure 22: Thought process), final decision (Figure 23: Final decision), and key points (Figure 21: Image from batch & key points), along with its evaluation of it

7. Conclusion/Discussion

This research successfully achieved its primary objective: the development of a functional prototype of a value-aware vacuum cleaner. The results demonstrate that AI can be meaningfully integrated into everyday appliances in a manner that respects and adapts to human values. The progress made throughout this project is encouraging, and the outcome affirms the potential of AI-driven, context-sensitive behavior in domestic robotics.

That said, the final product presents several opportunities for future improvement and expansion. Implementing real-time navigation and mapping would allow the vacuum cleaner to operate autonomously in more complex and dynamic environments. Incorporating a persistent, whole-house visual feed could enhance its situational awareness and enable more informed decision-making over time. Additionally, object recognition capabilities such as identifying and avoiding valuable or fragile items like jewelry would further increase the system's safety and practical utility.

Improvements to the Qt-based user interface could also enrich user interaction by offering deeper control, clearer feedback, and an overall smoother experience. As AI technologies continue to evolve rapidly, there is vast potential for further exploration and innovation within this domain.

This project provides a solid foundation for future developments in value-aware robotics, illustrating a promising direction for integrating ethical reasoning into everyday intelligent systems.

References

Abbo, G. A., & Belpaeme, T. (2023). Users' perspectives on value awareness in social robots. In *Proceedings of the 1st Workshop on Perspectives on Moral Agency in Human-Robot Interaction*. Presented at the HRI2023: the 18th ACM/IEEE International Conference on Human-Robot Interaction, Stockholm, Sweden. https://doi.org/10.5281/zenodo.8123742

Chen, Z., Li, Y., & Wang, K. (2024, July 16). *Optimizing reasoning abilities in large language models: A step-by-step approach*. TechRxiv. https://doi.org/10.36227/techrxiv.172114650.09447731/v1

Gabriel, I. (2020). Artificial intelligence, values, and alignment. *Minds and Machines, 30*, 411–437. https://doi.org/10.1007/s11023-020-09539-2

Gogoi, R.K. (2022) "A Software System for A Finite State Machine (FSM)," International Journal for Research in Applied Science and Engineering Technology, 10(7), pp. 795–806. https://doi:10.22214/ijraset.2022.44711

Guan, J., Dodge, J., Wadden, D., Huang, M., & Peng, H. (2024). Language models hallucinate, but may excel at fact verification. *Proceedings of NAACL 2024*. arXiv preprint arXiv:2310.14564. https://doi.org/10.48550/arXiv.2310.14564

Jing, J., Gao, T., Zhang, W., Gao, Y., & Sun, C. (2022). Image Feature Information Extraction for Interest Point Detection: A Comprehensive Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(4), 4694–4712. https://doi.org/10.1109/tpami.2022.3201185

Kipp, M. (2024). From GPT-3.5 to GPT-4.o: A leap in AI's medical exam performance. *Information, 15*(9), 543. https://doi.org/10.3390/info15090543

Krishna, A. K. M. J., Babu, A. V., Damodaran, S., James, R. K., Murshid, M., & Warrier, T. S. (2024). *ROS2-powered autonomous navigation for TurtleBot3: Integrating Nav2 stack in Gazebo, RViz and real-world environments*. In *Proceedings of the 2024 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)* (pp. 1–6). IEEE. https://doi.org/10.1109/SPICES62143.2024.10779642

Lee, M. (2023). A Mathematical Investigation of Hallucination and Creativity in GPT Models. Mathematics, 11(10), 2320–2320. https://doi.org/10.3390/math11102320

Li, H., Milani, S., Krishnamoorthy, V., Lewis, M., & Sycara, K. (2019). *Perceptions of domestic robots' normative behavior across cultures*. In Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (pp. 345–351). ACM. https://doi.org/10.1145/3306618.3314251

Liu, H., Ning, R., Teng, Z., Liu, J., Zhou, Q., & Zhang, Y. (2023). Evaluating the Logical Reasoning Ability of ChatGPT and GPT-4. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2304.03439

Luger, E., & Sellen, A. (2016). "Like having a really bad PA": The gulf between user expectation and experience of conversational agents. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 5286–5297. https://doi.org/10.1145/2858036.2858288

Mu, N., Lu, J., Lavery, M., & Wagner, D. (2025). A closer look at system prompt robustness. *arXiv*. https://doi.org/10.48550/arXiv.2502.12197

Pan, J., Yuan, Z., Zhang, X., & Chen, D. (2022, December). *YouHome system and dataset: Making your home know you better*. In 2022 IEEE International Symposium on Smart Electronic Systems (iSES) (pp. 414–420). IEEE. https://doi.org/10.1109/iSES54909.2022.00091

Parnami, A., & Lee, M. (2022). Learning from few examples: A summary of approaches to few-shot learning. *arXiv*. https://doi.org/10.48550/arXiv.2203.04291

Puck, L., Keller, P., Schnell, T., Plasberg, C., Tanev, A., & Heppner, G. (2021). *Performance evaluation of real-time ROS2 robotic control in a time-synchronized distributed network*. In *Proceedings of the 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)* (pp. 1670–1676). IEEE. https://doi.org/10.1109/CASE49439.2021.9551447

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., & Chadha, A. (2024, February 5). *A systematic survey of prompt engineering in large language models: Techniques and applications*. arXiv. https://doi.org/10.48550/arXiv.2402.07927

Snell, C., Yang, S., Fu, J., Su, Y., & Levine, S. (2022). Context-aware language modeling for goal-oriented dialogue systems. *Findings of the Association for Computational Linguistics: NAACL 2022*, 2351–2366. https://doi.org/10.18653/v1/2022.findings-naacl.181

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*, 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D. (2022, January). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv. https://doi.org/10.48550/arXiv.2201.11903

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023, February 21). *A prompt pattern catalog to enhance prompt engineering with ChatGPT*. arXiv. https://doi.org/10.48550/arXiv.2302.11382

Willman, J. (2021). Overview of PyQt5. In *Modern PyQt* (pp. 1–42). Apress. https://doi.org/10.1007/978-1-4842-6603-8_1

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., & Chi, E. (2023, April). *Least-to-Most Prompting Enables Complex Reasoning in Large Language Models*. arXiv. https://doi.org/10.48550/arXiv.2205.10625

Appendix

Camera Node
+ signal_emitter: QObject
+ node_name: string
+ image_received: bool
+ last_image_time: float
+ timer: float
+ camera_subscription: Image
+ camera_publisher: Image
+ camera_callback()
+ check_camera_status()
+ start_camera_service()
+ camera_service_response()

Figure 25: Table for Camera Node, this image shows a table of the Camera Node with its variables and functions

Robot Controle Node	
+ signal_emitter: QObject	+ r_close_to_object: bool
+ node_name: string	+ is_cleaning: bool
+ dock_status: bool	+ is_observing: bool
+ in_transition: bool	+ degrees: float
+ current_state: string	+ dock_status_subscription: DockStatus
+ time_wait: float	+ rosout_subscription: Log
+ state_received: bool	+ mode_subscription: String
+ fcl_close_to_object: bool	+ cmd_vel_punlisher: Twist
+ fcr_close_to_object: bool	+ hazard_detection_subscription: HazardDetectionVector
+ fl_close_to_object: bool	+ ir_intensity_subscription: IrIntensityVector
+ fr_close_to_object: bool	+ odom_subscribtion: Odometry
+ l_close_to_object: bool	
+ dock_status_callback()	+ stop_movement()
+ basic_nav_callback()	+ turn_around()
+ mode_callback()	+ play_sound()
+ hazard_detection_callback()	+ play_startup_sound_async()
+ ir_intensity_callback()	+ play_driving_sound_async()
+ odom_callback()	+ play_stop_sound_async()
+ nav_dock()	+ is_cleaning_sound_check()
+ dock_worker()	+ stop_current_sound()
+ nav_undock()	+ start_observation_sequence()
+ undock_worker()	+ observation_sequence()
+ mode_selection()	+ get_yaw_from_quaternion()
+ start_cleaning_sequence()	+ angle_diff()
+ cleaning_sequence()	+ stop_observation()
+ stop_cleaning()	

Figure 26: Table of Robot Control Node, this image shows a table of the Robot Control Node with its important variables and functions

Artificial Intelligence Node	
+ signal_emitter: QObject	+ iteration: int
+ node_name: str	+ conversational_history: array
+ nlp: Language	+ check_context: array
+ last_image_time: float	+ keypoints: array
+ temp_frames: array	+ thought_process: str
+ current_frames: array	+ final_decision: str
+ current_frames_time: dict	+ summarization_decision: str
+ function_context: str	+ image_folder_path: str
+ action_reflection: str	+ image_folder: str
+ keypoint_extraction_context: str	+ base64_image: str
+ cleaning_check_context: str	+ current_mode: str
+ gpt_page_names: dict	+ previous_mode: str
+ connected_page: str	+ cleaning_length: str
+ version_choices_names: dict	+ time_past_cleaning: str
+ version_choice: str	+ is_process_running: bool
+ time_context: str	+ allow_camera_intake: bool
+ observation_length_context: str	+ camera_subscribtion: Image
+ start_time_observation: str	+ mode_publisher: String
+ current_time_observation: str	
+ camera_footage_callback()	+ extract_keypoints_from_iteration()
+ get_image_folder()	+ value_aware_thought_process()
+ image_to_base64()	+ final_decision_process()
+ time_upkeep()	+ summarize_and_evaluate_decision_making()
+ get_keywords()	+ appand_conversational_history()
+ unwanted_characters()	+ check_if_cleaning_could_be_disturbed()
+ extract_mode()	+ start_camera_intake()
+ cleaning_timer()	+ stop_camera_intake()
+ start_AI_processing()	+ wait_for_all_footage()
+ main_loop_value_aware_process()	+ delete_frames()
+ reset_process()	

Figure 27: Table of Artificial Intelligence Node, this image shows a table of the Artificial Intelligence Node with its important variables and functions

CONTACT

Senne Lenaerts | student
r0938299.student@thomasmore.be
Tel. + 32 493 44 62 89
follow us on
www.thomasmore.be
fb.com/ThomasMoreBE
#WeAreMore

THOMAS MORE