Precision Farming Using Autonomous Vehicles: An Approach to the Design with CONSENS and MUML

Luis Fernando Rodriguez Gutierrez[1], Leander Hackmann[2], Hazhir Amiri[3], Furkan Iskender[4]

Abstract: The Smart AV project exemplifies the integration of advanced mechatronic principles with precision agriculture technology. The MUMI diagram illustrates a sophisticated system architecture designed for autonomous agricultural vehicles, capable of disease detection, weed control, and pesticide application. At its core, the Communication Module coordinates data flow between various interfaces, such as Disease Position, Protection Command, and Weed Status Report. The Main Controller Unit serves as the system's brain, processing inputs from the Computer Vision Unit and Camera, which utilize image recognition to classify field conditions. This information guides the Electric Motor and Propeller subsystems to execute precise kinematic actions, such as lifting and thrusting, while the Weeder Motor and Blades are engaged for mechanical weed removal. The Pesticide Actuator subsystem ensures targeted pesticide dispensing, enhancing efficiency and reducing environmental impact. Overall, this UML design highlights the seamless integration of mechanical, electronic, and software components in creating a highly autonomous and effective agricultural solution.

1 Introduction

Precision farming is an innovative approach that leverages advanced technologies to optimize agricultural practices. In this project, we focus on developing an autonomous robot designed to navigate a scaled-down road track and identify various plant types, such as healthy and diseased sugar beets and weeds, using a single camera sensor. The primary objectives are to enhance efficiency in plant identification and autonomous navigation. By improving these areas, we aim to boost overall agricultural productivity, promote ecological sustainability, and increase economic efficiency. This project represents a significant step towards smarter and more sustainable farming practices, addressing the growing global need for food production through advanced robotics and precision agriculture techniques.

1.1 Motivation

The motivation behind this project is to enhance the efficiency and sustainability of agricultural practices through the integration of advanced mechatronic systems and autonomous

[1] 7219085, luis.rodriguez001@stud.fh-dortmund.de, Master's student, FH Dortmund
[2] 7217912, leander.hackmann001@stud.fh-dortmund.de, Master's student, FH Dortmund
[3] 7218509, hazhir.amiri001@stud.fh-dortmund.de, Master's student, FH Dortmund
[4] 7219067, furkan.iskender001@stud.fh-dortmund.de, Master's student, FH Dortmund

vehicles. By leveraging precision agriculture techniques, the project aims to address challenges such as disease detection, weed control, and targeted pesticide application.

1.2 Methodology

To tackle the design of a solution for the precision farming problem, a methodology must consider all aspects of the problem. This includes addressing the general and technical requirements of the project, creating a visual design that satisfies high-level stakeholders and paves the way for future development, and developing technical specifications for actual production. For this purpose, the CONSES specification technique was employed. Subsequently, the system was modeled using Mechatronic UML to provide a detailed technical representation, facilitating the development of a comprehensive technical architecture.

1.3 Goals

The goal is to create a functional prototype that fulfills the stakeholders' requirements by developing a sophisticated system architecture for autonomous agricultural vehicles. This includes creating a communication module for seamless data flow, implementing advanced image recognition for field condition classification, designing kinematic actions for precise mechanical operations, and ensuring targeted pesticide dispensing to minimize environmental impact.

2 CONSENS

Using the CONSENS technique, different aspects of the system can be semi-formally specified and considered, thus reducing the system's overall complexity. Furthermore, it provides a medium that both non-technical stakeholders and engineers can understand well enough. At first, the requirements of the system were drafted to allow considering the limitations and the scope of the project for further specification development of other aspects of the system. After the requirements reached a stable level where different objects could be agreed upon, the other facets of the system, such as functions, behaviors, and active structures were specified.

2.1 Requirements

2.1.1 High-Level Requirements

The development of an autonomous drone with a microcontroller involves several key categories: hardware, software, functional, testing, and documentation requirements. Hardware needs include a video-capable micro-controller module, power supply, Wireless communication module, ventilated casing, two high-resolution cameras, drone frame, high-efficiency motors, motor driver, battery pack, propellers, wiring, breadboard, and mechanisms for weed elimination and fertilizer distribution. Software requirements encompass a Linux-based OS, Python or C++ development, OpenCV, TensorFlow or PyTorch, ROS, and Jetson Inference. Functional requirements cover autonomous navigation, lane detection, plant identification and treatment, and sign recognition. Testing involves performance metrics in both simulations and real-world scenarios. Documentation demands comprehensive user and technical manuals and a version-controlled repository for source code with thorough comments and documentation.

The high-level requirements for the Smart AV project are designed to ensure the system's functionality, efficiency, and reliability in an agricultural setting. These requirements are as follows:

- **Requirement 1: Detection, Identification, and Elimination of Weeds**
  - The system shall utilize advanced image recognition technologies to detect and identify different types of weeds present in the agricultural field.
  - The system should be capable of processing images in real-time to facilitate immediate action.
  - The identification process must be accurate to distinguish between crops and weeds to avoid any damage to the crops.
  - The system shall eliminate identified weeds in a safe manner that may not harm the crops.
- **Requirement 2: Detection, Identification, and Protection of Unhealthy Plants**
  - The system shall utilize advanced image recognition technologies to detect and identify unhealthy plants, differentiating them from healthy plants.
  - The system shall apply pesticides specifically to the identified unhealthy plants, minimizing the use of chemicals and reducing environmental impact.
  - The system should allow for adjustable dosage control based on the type and severity of the diseases present in the unhealthy plant.
- **Requirement 3: Autonomous Navigation**

  - The system shall navigate autonomously through the agricultural field without human intervention.
  - It must be able to detect and avoid obstacles, ensuring safe operation in a dynamic environment.
  - The navigation system should include route planning to optimize the coverage area and efficiency of operations.

- **Requirement 4: Data Collection and Reporting**
  - The system shall collect data on field conditions, including soil moisture, temperature, and crop health.
  - This data should be reported to a central system or user interface for analysis and decision-making.
  - The reporting system must be user-friendly and provide actionable insights to improve agricultural practices.

- **Requirement 5: Operational Continuity**
  - The system shall be capable of continuous operation for at least 8 hours to cover large fields without frequent interruptions.
  - It must have a reliable power source, such as a high-capacity battery or an efficient energy management system.
  - The system should include mechanisms for easy maintenance and quick troubleshooting to minimize downtime.

2.2 Environment

The environment model details the interactions between the autonomous vehicle (AV) and its surrounding elements. At the heart of this system is the AV, which operates in a complex environment influenced by various factors, including signs, environmental conditions, path marks, weeds, and unhealthy plants. Figure 1 depicts the environment model as a diagram where the AV interacts with outside elements.

Signs provide crucial information to the AV by indicating different sign types that the vehicle must recognize and interpret to navigate correctly and perform tasks accurately. These signs ensure the AV can adapt to various instructions and changes along the path it needs to take.

Environmental conditions such as snow, rain, and wind play a significant role in the robot's operation. These conditions affect both the AV and the path it follows. The environmental feedback helps the AV adjust its operations to maintain efficiency and accuracy in diverse weather scenarios. As the AV uses a wide-angle camera to interpret its environment, weather events can affect the data received from the camera, thus impacting its accuracy in path and

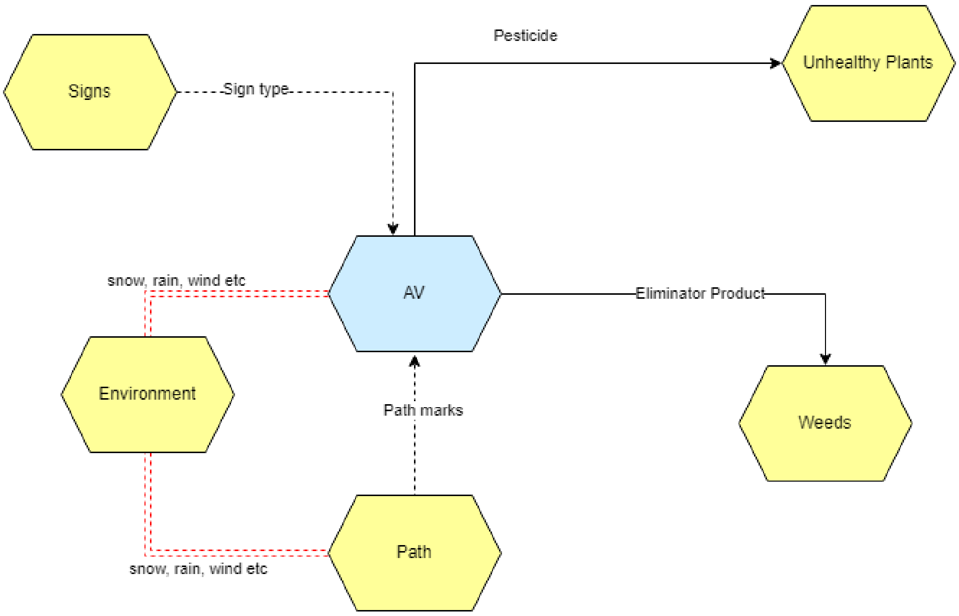

Fig. 1: CONSENS Environment model of the AV

plant detection. By specifying and taking note of possible weather conditions, measures can be put into place to improve data quality. Concerning efficiency, weather events could negatively impact roads and the path the AV needs to take; by reacting to these conditions, AV could recalculate and take a new path to reduce fuel usage.

The path provides the AV with path marks, guiding its navigation and ensuring it stays on course while performing its agricultural tasks.

The AV's primary functions involve managing weeds and unhealthy plants. The AV uses an eliminator product to target and remove weeds, ensuring the crops do not compete for resources. Additionally, it applies pesticides to unhealthy plants to protect and treat them, promoting healthier crop growth. Overall, the environment model illustrates a sophisticated interaction network where the AV is central, receiving inputs from signs, environmental conditions, and path marks, while outputting actions such as weed elimination and pesticide application. This integration ensures that the autonomous robot can navigate its environment effectively and maintain optimal agricultural productivity.

2.3 Application Scenarios

The model considers two scenarios that the smart AV is very likely to encounter and has to respond to. Firstly, as depicted in figure 2, the AV could encounter weeds and unhealthy plants along the path it's taking and would have to react accordingly.

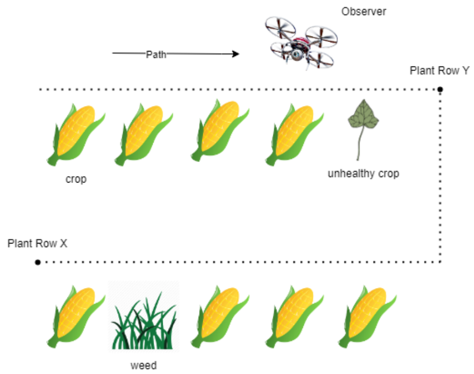

Fig. 2: CONSENS application scenario regarding different roles of the AV

To successfully identify weeds and unhealthy plants and apply the relevant products, the AV would have to correctly classify the image data and then take on the role of a weeder or crop protector to complete these tasks.

In the second scenario, AV number 1 comes across several weed plants that it needs to eliminate. To expedite the process, the AV communicates with other AVs (in this scenario, AV number 2) to carry out the task together. Figure 3 depicts the visual representation of this scenario.

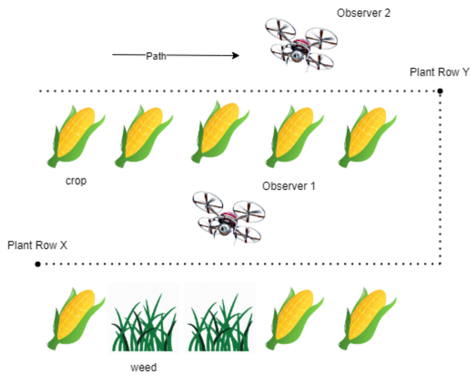

Fig. 3: CONSENS application scenario regarding communication between AVs

2.4 Functions

This aspect of the CONSENS technique provides a way to specify the functionality of the system hierarchically. The functions are tasks the system has to execute to fulfill its goal. For this, a top-down hierarchical design was considered where top functions (tasks) were broken down into smaller functions that the system carries out to fulfill the top tasks. This provides a straightforward path to identifying the system's top functions and breaking them down into their building blocks which in turn helps to understand everything the system needs to undertake to fulfill our end goals.

The system functions of the smart AV are designed to ensure effective agricultural maintenance and autonomous navigation. For agricultural maintenance, the primary goal is to maintain crops and eliminate weeds. Figure 4 depicts the breakdown of these tasks in a top-down hierarchical manner.

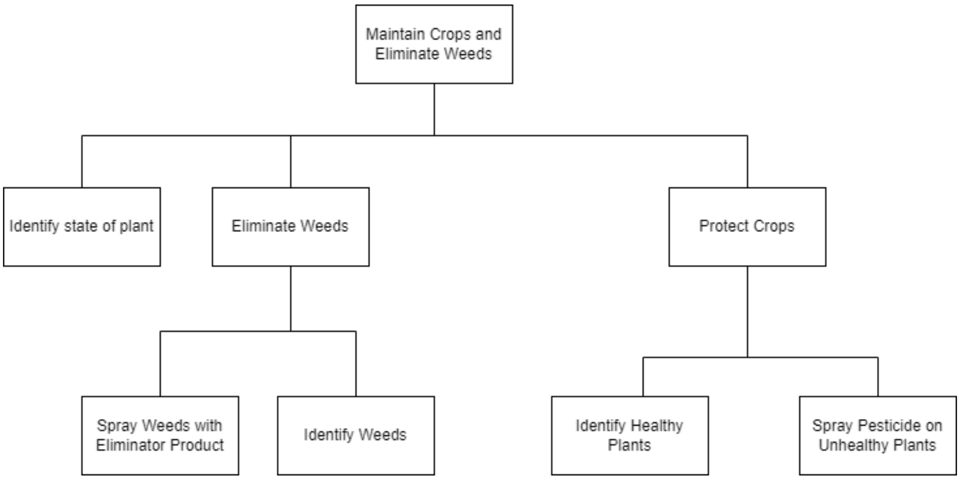

Fig. 4: CONSENS Functions diagram for eliminating weeds and maintaining crops

The task of eliminating weeds and maintaining crops involves several critical functions. First, the system identifies the state of each plant, determining whether they are healthy or unhealthy. For weed elimination, the robot identifies weeds present among the crops and subsequently sprays them with an eliminator product to remove them and prevent resource competition. To protect crops, the system identifies healthy plants, ensuring they remain untreated, and sprays pesticides only on unhealthy plants to improve their health and prevent disease spread.

On the navigation side, the robot's primary objective is to drive autonomously around the track. This involves adjusting speed and direction based on the environment and detected pathways. Figure 5 in turn shows the breakdown of said tasks.

Speed adjustments are managed through accelerating and decelerating systems, while

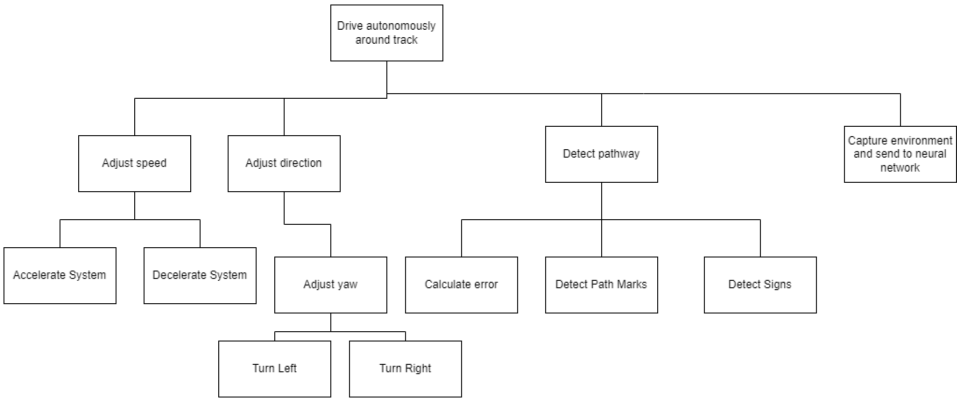

Fig. 5: CONSENS Functions of AV navigation

direction adjustments include altering yaw and making turns. The system also detects pathways by calculating errors, detecting path marks, and recognizing signs. Additionally, it captures environmental data, which is sent to a neural network for further processing. This comprehensive functionality ensures that the robot can navigate independently while performing its agricultural tasks.

2.5 Active Structure

The active structure of the AV describes the system's elements, their attributes, and the relationships between these elements, focusing on specifying the system's architecture. The system is composed of two primary subsystems: the Driving System and the Plant System. This is depicted in figure 6.

The Vision System collects data from the Observer (abstracted to be any sort of input, but mainly in this project a high-resolution camera), does some initial processing, and sends relevant data to responsible sub-systems. The Driving System receives vital path data from the Vision System path data, which is processed by the Software Unit. This unit communicates with the Control Unit, which uses the data to adjust the robot's throttle and yaw, directing voltage to the Stepper Motor for precise movements and the DC Motor for driving force.

The Plant System also receives data on weeds and crops. This data is further processed by the Software Unit of the Plant System, which identifies weeds and unhealthy crops. The Weeder component receives weed data and applies an eliminator product to remove the weeds. On the other hand, the Crop Protector receives data on unhealthy crops and sprays pesticides to protect and treat them.

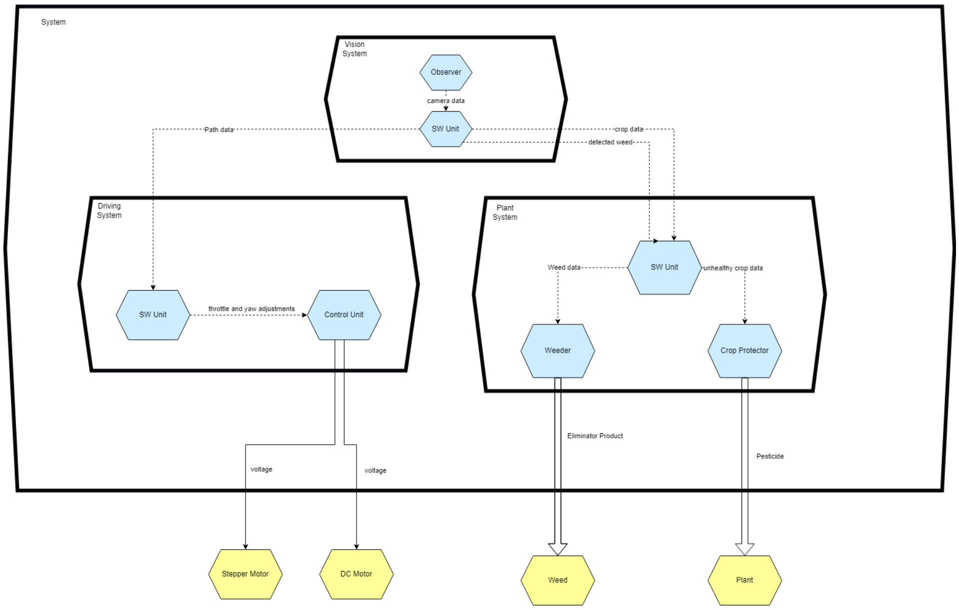

Fig. 6: CONSENS Active Structure diagram of the AV

Together, these subsystems enable the autonomous robot to navigate effectively and perform targeted agricultural tasks, ensuring proper crop maintenance and weed elimination. This integrated structure ensures that the robot is able to navigate autonomously while it carries out its agricultural tasks. Furthermore, these recognized systems closely resembled or are identically specified further in MUML diagrams and the behavioral model. For example, the Observer component within the MUML specification represents the Vision System as specified within the CONSENS models.

2.6 Shape

The CONSENS shape model is meant to provide a visual representation of the project that serves as a visual guide for both technical and non-technical stakeholders. Non-technical stakeholders can use this to have an idea of what the end product could look like and also use it as an end-product representation for fundraising. On the other hand, the shape model helps mechanical and system architects identify the needs of the end product. As a result, figure 7 was designed to reflect these goals.

textwidthtextwidth

Fig. 7: CONSENS shape of the AV

The shape represents a cutting-edge, autonomous agricultural drone designed for efficient navigation in agricultural fields. Its sleek, modern design features a blue-and-white color scheme. At its center, a robust antenna ensures reliable communication for remote operation and data transmission. The drone's four propellers, attached to each arm, provide lift and stability for precise maneuvering over various terrain types. A high-resolution camera mounted at the front of the drone enables visual identification of weeds and unhealthy crops, while a pesticide/protector product nozzle at the sides can dispense targeted products. The sturdy landing gear ensures safe and stable landings on flat surfaces. This innovative design combines standard drone design principles with project-specific needs to minimize costs (by saving time and using a standard design) and deliver expectations at the same time with the additions.

3 Mechatronic UML Specification

3.1 Component Interaction Diagram

The Component Interaction Diagram illustrates the complex interactions between various modules within the Smart AV system. It highlights the communication and coordination necessary for effective precision agriculture. The key modules in this diagram are the Observer, Crop Protection, and Weeder, each interfacing with the environment and contributing to the system's overall functionality.

The Observer module is responsible for gathering and processing visual data, which is critical for identifying and managing agricultural tasks. It represents the Vision System from the CONSENS models. It includes several ports:

3.1.1 Observer: Smart AV

- **ActionPositionPort**: Handles positional commands for actions, enabling precise movements and operations.
- **CoordinationPort**: Coordinates activities between different system components, ensuring synchronized operations.
- **ReportRequestPort**: Manages requests for data reports, facilitating timely information retrieval.
- **KinematicActuation**: Manages kinematic actions to interact with the environment physically.
- **VisualInputPort**: Receives visual input for processing, essential for monitoring and decision-making.

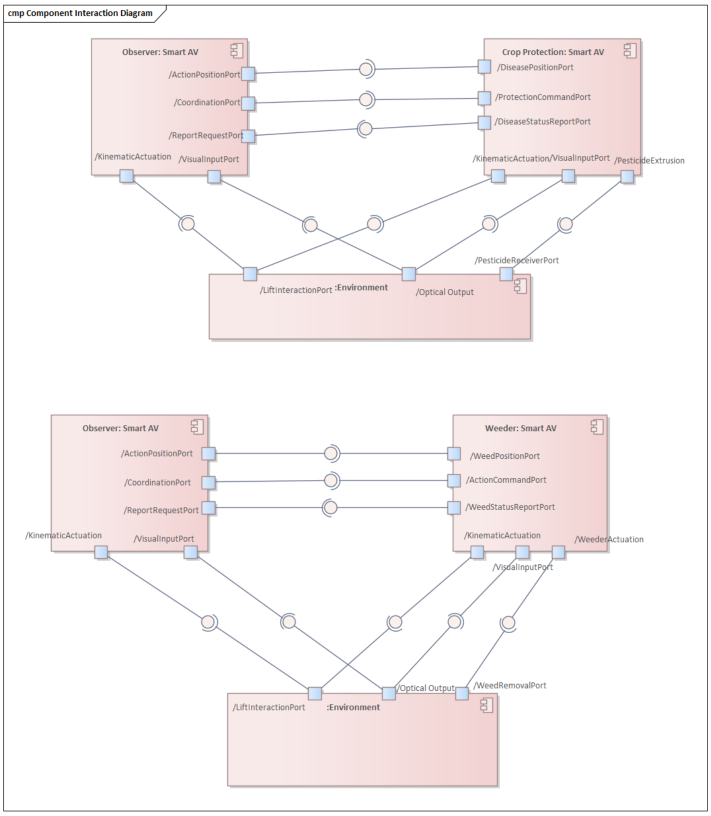

Fig. 8: Component Interaction Diagram

The Crop Protection module, a sub-element of the Plant System within the CONSENS active structure model, is designed to manage and execute tasks related to protecting crops from diseases and pests. It includes ports such as:

3.1.2 Crop Protection: Smart AV

- **DiseasePositionPort**: Handles the positioning related to disease detection, ensuring accurate targeting.
- **ProtectionCommandPort**: Issues commands for protection mechanisms, activating necessary responses.
- **DiseaseStatusReportPort**: Reports the status of detected diseases, providing essential feedback for continuous monitoring.

- **PesticideExtrusion**: Controls the extrusion of pesticides, ensuring precise application to targeted areas.

The Weeder module, another sub-element of the Plant System within the CONSENS active structure model, is tasked with the identification and removal of weeds. It operates through ports such as:

3.1.3 Weeder: Smart AV

- **WeedPositionPort**: Manages the positioning related to weed detection, enabling accurate identification and targeting.
- **ActionCommandPort**: Issues commands for weeding actions, ensuring effective weed management.
- **WeedStatusReportPort**: Reports the status of weed management, providing data for assessment and adjustment.
- **WeederActuation**: Controls the actuation of the weeder mechanism, ensuring physical interaction with the environment for weed removal.
- **WeedRemovalPort**: Manages the removal of weeds, ensuring that the field is kept clear of unwanted plants.

3.2 Starter Component Diagram

The Starter Component Diagram provides a detailed overview of the system's architecture, showcasing the interactions and relationships between various subsystems. This diagram is essential for understanding the foundational structure of the Smart AV system, including its core components and their interactions. This model also points out the lack of a communication module within the CONSENS models; a module which is needed to ensure proper communication within sub-systems in an AV and also with other AVs.

The Communication Module plays a pivotal role in facilitating data exchange and coordination among different subsystems. It includes ports like:

3.2.1 Communication Module

- **ReportRequestPort**: Manages requests for reports, ensuring that necessary data is collected and communicated efficiently.
- **CoordinationPort**: Coordinates communication between modules, enhancing system integration and performance.

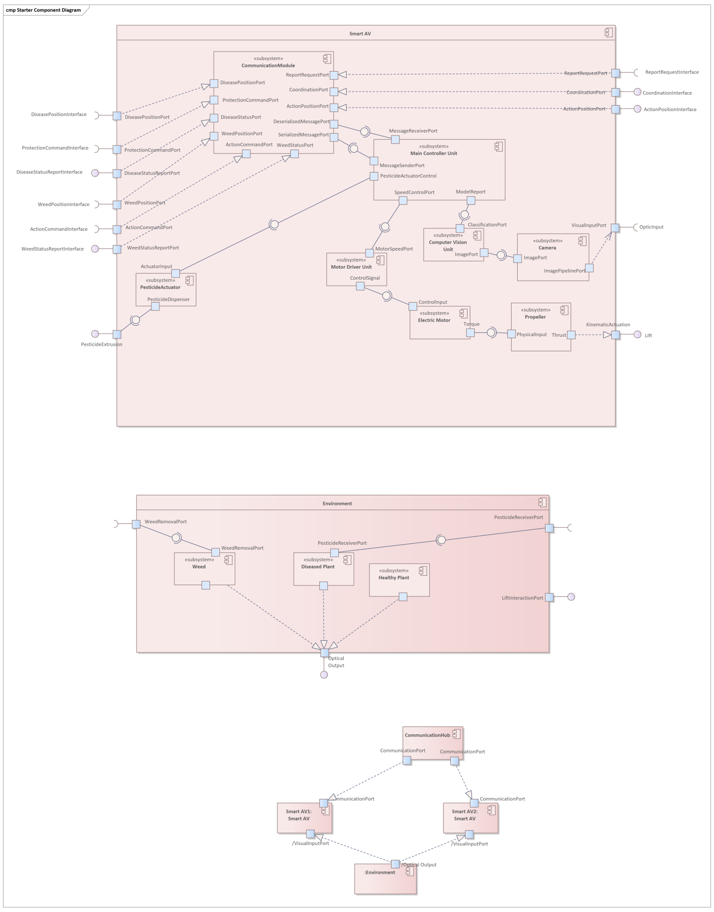

Fig. 9: Starter Component Diagram

- **ActuatorPositionPort**: Handles position data for actuators, ensuring precise movements and actions.

The Main Controller Unit is the brain of the Smart AV system, managing inputs and outputs across various components. It features ports such as:

3.2.2 Main Controller Unit

- **MessageReceiverPort**: Receives messages from various components, facilitating communication and coordination.

- **MotorSpeedPort**: Controls motor speed, ensuring that movements are performed at the correct pace.
- **ImageInputPort**: Receives image data for processing, which is crucial for visual monitoring and decision-making.
- **KinematicActuation**: Controls kinematic actions, enabling physical interactions with the environment.

The Pesticide Acutator is responsible for applying eliminator products to weeds or protector products to unhealthy crops. It includes:

3.2.3 Pesticide Actuator

- **PesticideDispenser**: Manages the spraying of pesticide/protector products using the spray arm.

The Environment Interaction subsystems manage the interaction of the Smart AV system with its surroundings. This includes:

3.2.4 Environment Interaction

- **WeedRemovalPort**: Manages the removal of weeds from the environment, ensuring a clear and healthy field.
- **PesticideReceiverPort**: Receives pesticides for application, ensuring that the correct amount is dispensed.
- **OpticalOutput**: Manages optical output for monitoring, providing visual feedback on field conditions.

The diagram also shows interactions between multiple Smart AV units, demonstrating how they communicate and coordinate with each other to achieve common goals:

3.2.5 Smart AV: Interaction Between Multiple Units

The diagram includes several Smart AV units communicating through dedicated communication ports and interacting with the environment. This highlights the collaborative aspect of the system, ensuring comprehensive coverage and efficiency.

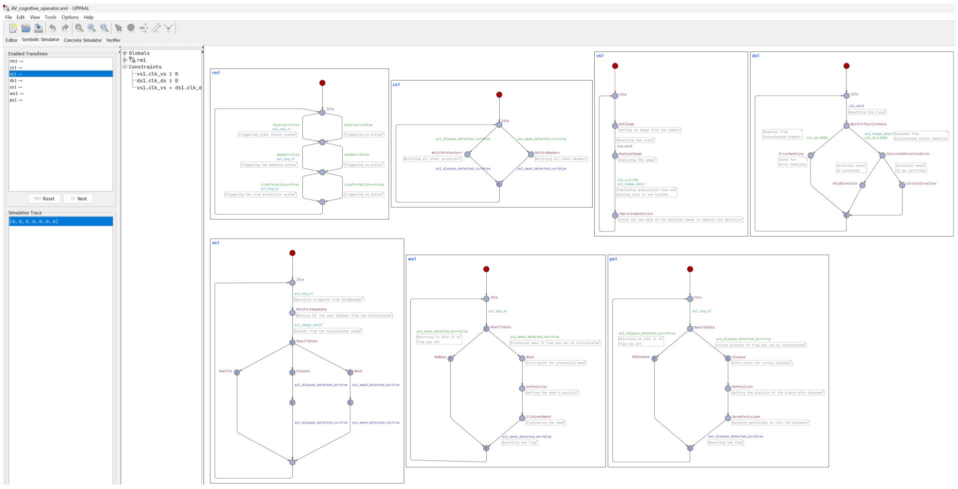

Fig. 10: Subsystems of the Cognitive Operator

4 Cognitive Behavior

The Cognitive Operator and therefore the Cognitive Behavior of the AV was modeled by using a set of intercommunicating timed automata, representing different sub-systems. The whole Cognitive Operator consists of seven sub-systems (RoleManager, VisionSystem, Driving System, Status System, WeedingSystem, CropProtectionSystem, and CommunicationSystem), each implementing one of the desired functionalities of the Cognitive Operator. The following functionalities were implemented:

- Enabling the AV to be in one or multiple functional roles (roles: "Plant Observer", "Weeder", "Crop Protection").
- Obtaining images, processing the information, and sending it to the different subsystems while trying to optimize the detection continuously.
- Observing if the direction of movement and the position of the AV is correct or if it needs to adjust its direction by the information of the camera.
- When in role "Plant Observer": Categorizing and determining the status (healthy, diseased, weed) of the detected plants and sending it to the different subsystems as well to other AVs.
- When in role "Weeder": Getting the exact position of the weeds and eliminating them.
- When in role "Crop Protection": Getting the exact position of the diseased plants and applying pesticides or other treatments.

The tool UPPAAL, a modeling tool with integrated validation and verification for networks of timed automata, was used for building, simulating, and checking the system.

4.1 VisionSystem

The sub-system VisionSystem captures images from the optical components and sensors of the AV, analyzes the image, and passes the data to the sub-systems DrivingSystem and StatusSystem while continuously trying to improve the detection. This behavior is modeled by four sequential states. In the first state, an image is obtained from the optical components and sensors. It is then analyzed in the state AnalyzeImage. As analyzing an image takes a certain time in reality, a delay of 100 ticks was implemented by resetting an internal clock when entering AnalyzeImage and applying a greater or equal to 100 guards at the transition to the next state. In this transition, the channel image_data also gets sent to simulate the data passing to the sub-systems. In the following state, ImprovingDetection, the continuous optimization of the detection is modeled. After this state, the system proceeds to the idle state to start the next cycle.

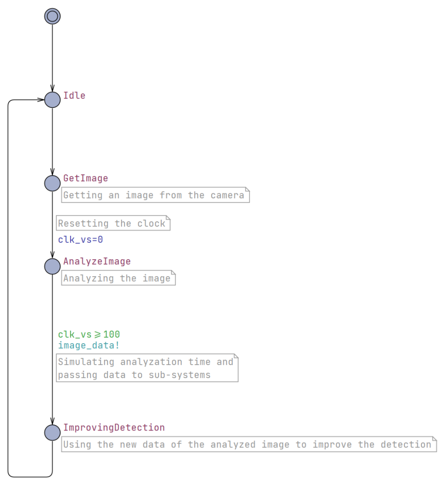

Fig. 11: Timed automaton of the VisionSystem

4.2 DrivingSystem

The sub-system DrivingSystem observes if the direction of movement and the position of the AV is correct or if it needs to adjust its direction. This is done by using the information of the optical components and sensors that were obtained by the VisionSystem. After leaving the state Idle and moving into the state WaitForPositionData, the sub-system waits for the data of the VisionSystem. This is implemented by awaiting the signal position_data, which is connected to the signal image_data of the VisionSystem in the SystemDeclarations. As the continuous observation of AV’s movement is probably the most safety-critical part and is therefore heavily time-constrained, a hard deadline for the arrival of the data was implemented by resetting an internal clock when moving to WaitForPositionData and checking its value when moving to the next states, ErrorHandling or CalculateDirectionError. If the deadline is exceeded (clock greater than 2500 ticks), the sub-system goes into the state ErrorHandling, where a functionality to ensure safe operation of the AV is executed before it proceeds to the state Idle to start the next cycle. If the deadline is not violated, the sub-system takes the transition from WaitForPositionData to CalculateDirectionError, where an error is calculated to decide whether the direction of the AV needs to be adjusted or not. If no adjustment is needed the transition to HoldDirection is taken and the sub-system proceeds to the state Idle to start the next cycle. If adjustment is needed, it proceeds to the state CorrectDirection to execute the correction of the AV's direction. After the correction, it proceeds to the state Idle to start the next cycle.

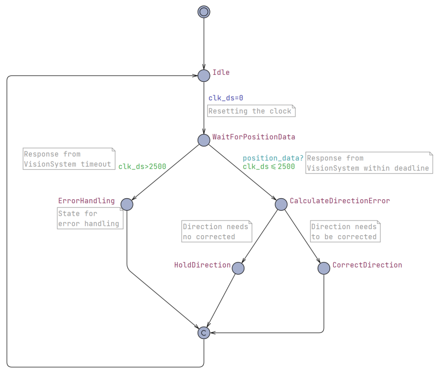

Fig. 12: Timed automaton of the DrivingSystem

4.3 RoleManager

The sub-system RoleManager takes care of the execution of the specific sub-systems that implement the desired functionalities when the AV is in the role of a Plant Observer, Weeder, or Crop Protection (or a combination of those three roles). As the roles of an AV can change during its runtime, the RoleManager checks periodically whether a role is currently active or not and sends a signal to the corresponding sub-system (StatusSystem, WeedingSystem, CropProtectionSystem) to trigger its operation. This is implemented by a timed automation with a set of four sequential states that runs in cycles. Between two states there are transitions with mutually exclusive guards where a flag (observer, weeder, cropProtection) is checked to be true or false. In the transitions with the check for the flag to be true, a signal (msg_s, msg_w, msg_p) to the corresponding sub-system is sent to trigger its operation.

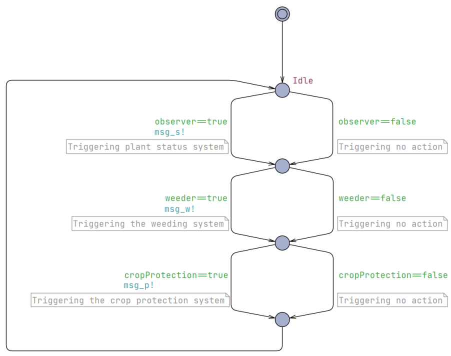

Fig. 13: Timed automaton of the RoleManager

4.4 StatusSystem

The sub-system StatusSystem implements the functionalities of the role Plant Observer. The operation of the system gets triggered by the RoleManger when receiving the signal trig_operation, which is connected to msg_s of the RoleManger in the SystemDeclarations. It then proceeds from the state Idle to the state WaitForImageData, where it waits for the next dataset of the VisionSystem. The transition to the next state, ReactToData, is triggered by the VisionSystem via the signal image_data, which indicates that the next dataset is available. In the state ReactToData, the data is processed and the actions that are needed to react to a certain situation, are executed. If a healthy plant is detected, no actions are needed and the system directly proceeds to the state Idle to wait for the trigger to start the next cycle. If a diseased plant or weed is detected, two flags (disease_detected_ps and disease_detected_cs or weed_detected_ws) and weed_detected_cs) are set and the system proceeds to the state Idle to wait for the trigger to start the next cycle. These flags are used to model the asynchronous notification of the corresponding internal sub-systems WeedingSystem or CropProtectionSystem, and the CommunicationSystem.

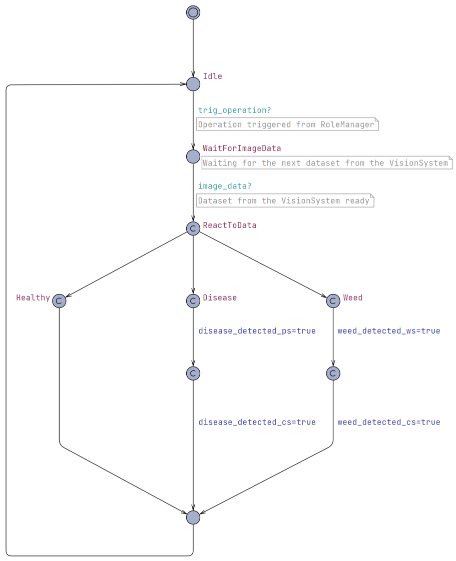

Fig. 14: Timed automaton of the StatusSystem

4.5 WeedingSystem

The sub-system WeedingSystem implements the functionalities of the role Weeder. Its structure is similar to the structure of the CropProtectionSystem. The operation of the system gets triggered by the RoleManger when receiving the signal trig_operation, which is connected to msg_w of the RoleManger in the SystemDeclarations. It then proceeds from the state Idle to the state ReactToData, where it checks if information about a plant that was categorized as weed is available. This is done by checking the flag weed_detected, which gets set by the StatusSystem and is connected to its variable weed_detected_ws in the SystemDeclarations. If the flag is not set, the sub-system directly proceeds via the state NoWeed to the state Idle to wait for the trigger to start the next cycle. If the flag is set, it proceeds to the state GetPosition via the state Weed, where the exact position of the weed is determined. After getting the exact position, it moves to the state EliminateWeed to take out the weed. It then proceeds to the state Idle while resetting the flag weed_detected and waits for the trigger to start the next cycle.
In the simulation, the flag to model available information about a plant that was classified as weed only gets set by the StatusSystem. In the real scenario, it also can be set by the CommunicationSystem when a piece of information from another AV is received. As the implementation of a simulation compromising several AVs extends the capabilities of UPPAAL, this feature was left out.

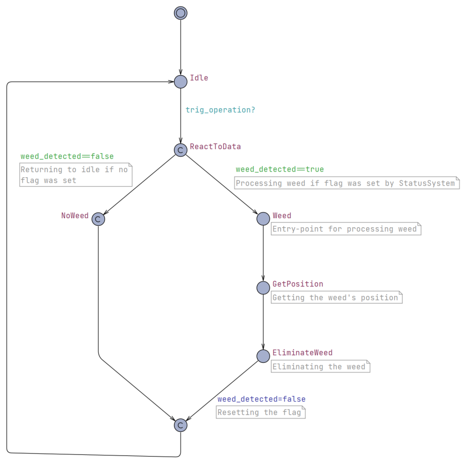

Fig. 15: Timed automaton of the WeedingSystem

4.6 CropProtectionSystem

The sub-system CropProtectionSystem implements the functionalities of the role Crop Protection. Its structure is similar to the structure of the WeedingSystem. The operation of the system gets triggered by the RoleManger when receiving the signal trig_operation, which is connected to msg_p of the RoleManger in the SystemDeclarations. It then proceeds from the state Idle to the state ReactToData, where it checks if information about a plant that was categorized as diseased is available. This is done by checking the flag disease_detected, which gets set by the StatusSystem and is connected to its variable disease_detected_ps in the SystemDeclarations. If the flag is not set, the sub-system directly proceeds via the state NoDisease to the state Idle to wait for the trigger to start the next cycle. If the flag is set, it proceeds to the state GetPosition via the state Disease, where the exact position of the diseased plant is determined. After getting the exact position, it moves to the state SprayPesticides to apply the treatment. It then proceeds to the state Idle while resetting the flag disease_detected and waits for the trigger to start the next cycle.
In the simulation, the flag to model available information about a diseased plant only gets set by the StatusSystem. In the real scenario, it also can be set by the CommunicationSystem when a piece of information from another AV is received. As the implementation of a simulation compromising several AVs extends the capabilities of UPPAAL, this feature was left out.

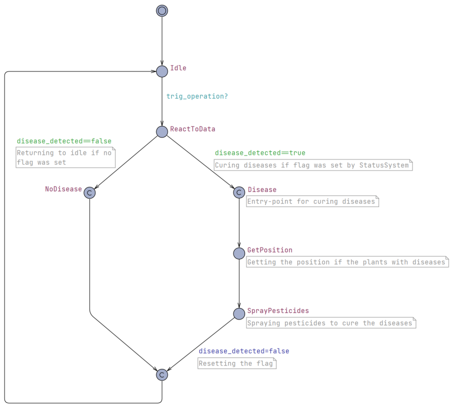

Fig. 16: Timed automaton of the CropProtectionSystem

4.7 CommunicationSystem

The sub-system CommunicationSystem implements the feature of notifying other AVs in the role of Weeder or Crop Protection if plants classified as weed or diseased plants were detected. This was modeled by an automaton that checks in cycles whether the flag disease_detected or weed_detected was set by the StatusSystem and then sends out messages to the other AVs in the corresponding states NotifyWeeders and NotifyProtectors. The flags are connected accordingly to the variables weed_detected_cs and disease_detected_cs of the StatusSystem and get reset after the other AVs are notified and the automaton proceeds to the state Idle.

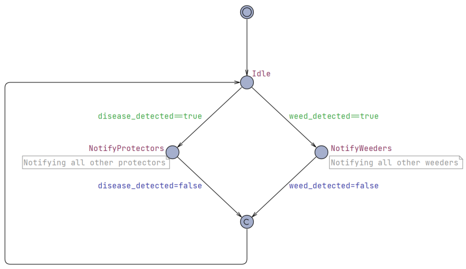

Fig. 17: Timed automaton of the CommunicationSystem

4.8 Verification of the model

As the Cognitive Operator compromises several synchronously and asynchronously interconnected sub-systems that are modeled by timed automatas, a verification of the system is needed to guarantee its correct operation. UPPAAL offers the possibility to implement custom checks of the system by its built-in verifier.
The following checks were implemented:

- *A[] not deadlock* : Checking if the system is free of deadlocks.
- *A[] (not (ds1.HoldDirection and ds1.CorrectDirection))*: Checking that the DrivingSystem is not holding and correcting the AV's direction at the same time.
- *A[] (vs1.AnalyzeImage imply vs1.clk_vs < 100)*: Checking whether the state AnalyzeImage of the VisionSystem is only active when the value of clk_vs is smaller than 100 clock ticks.
- *E<> ds1.ErrorHandling*: Checking if the state ErrorHandling of the DrivingSystem is reachable.
- *E<> ds1.CalculateDirectionError*: Checking if the state CalculateDirectionError of the DrivingSystem is reachable.

All checks resulted in a positive result, indicating a basic level of safety of the system. Especially the checks for a deadlock-free system and that the DrivingSystem is able to go into the error handling state and not correcting and holding the direction at the same time are the most important ones. The checks themselves and their amount should be adapted as the whole system is more specified.

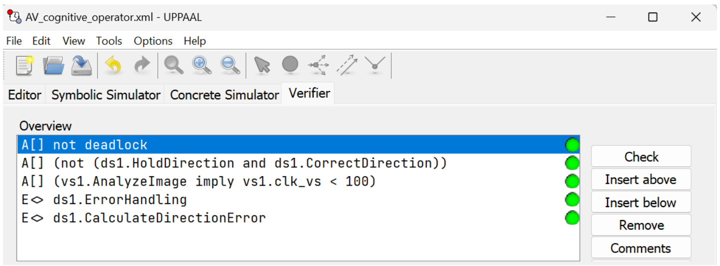

Fig. 18: Screenshot of the results of the implemented checks

5 Prototype Design and Components

The prototype of the autonomous agricultural vehicle (AV) is designed to incorporate advanced technologies for precision farming. The vehicle features a sleek and modern design with robust construction to withstand various field conditions. The following sections detail the key components and their functions.

Fig. 19: Prototype of the Autonomous Agricultural Vehicle (AV). Note: This image is AI-generated.

5.1 High-Resolution Cameras

The drone is equipped with high-resolution cameras mounted on top. These cameras are used for visual input and image processing, allowing the system to detect and identify weeds, crops, and other relevant field data.

5.2 Sensors

A complex array of sensors is located at the front of the drone for obstacle detection and navigation. These sensors ensure the drone can operate autonomously, avoiding obstacles and navigating through the field efficiently.

5.3 Pesticide Dispensing Unit

On one side of the drone, there is a precise pesticide dispensing unit. This unit is designed to apply pesticides directly to identified weeds or protector products to unhealthy crops, minimizing the use of chemicals and protecting the surrounding crops.

5.4 Propellers

The drone is equipped with high-efficiency propellers suitable for various flight conditions. These propellers provide the necessary lift and stability for the drone to operate effectively in different field conditions.

5.5 Main Controller and Communication Modules

The drone houses a compact unit containing the main controller and communication modules. Visible antennas ensure effective data transmission, allowing the drone to communicate with other systems and devices.

5.6 Safety Markings

The drone is painted in a professional, dark green color with reflective safety markings. These markings enhance the visibility of the drone, ensuring it can be easily seen by operators and other machinery in the field.

6 Conclusion

The smart AV project presents a seamless integration of various mechatronics principles to realize an innovative approach to precision farming. In the project's scope, an autonomous vehicle navigates through agricultural fields to carry out tasks that promote plant health and minimize the risk of damages. The AV achieves this by performing two main functions: identifying and eliminating weeds through image recognition techniques, and identifying unhealthy plants and using healing products to improve their health. By improving crop life in this way, higher agricultural yields will be achieved, which in turn help meet the ever-increasing food demands and ecological and economic expectations.

To successfully bring the idea of the smart AV to life, a methodology needed to be considered that would provide a clear overview for designing the end product and reduce the complexities that come with it. The CONSENS methodology for mechatronics provides just the right tools for such a feat. By considering different aspects of the same product, such as high-level requirements, technical functionalities and visual shape, this methodology paves the way for the later phases of the project. As a result, a product with the following main components was designed: a controller component as the brains; a vision component concerned with taking and the classifications of images; a weeder component as the responsible sub-system for identifying and eliminating weeds; a crop protector in charge of identifying unhealthy plants and treating them; a navigation component for autonomous path detection and navigation; and a communication module responsible for maintaining communication within the sub-systems and also with the other AVs. These components are not only decomposed into further detail within CONSENS diagrams themselves but also on a more detailed level within the MUML specifications and the behavioral model. Doing so, a seamless link between different models of the system is established and these models in turn support the specification of the components in detail.

In conclusion, the smart AV project embodies an innovative approach towards precision farming with the aim of improving ecological expectations and economic growth by promoting crop life. To tackle the complexities of designing such a project, the CONSENS methodology was utilized to design a high-level model of the program which was specified in technical detail by the respective MUML and behavioral models.

Affidavit

We, Hazhir Amiri, Luis Fernando Rodriguez Gutierrez, Leander Hackmann, and Furkan Iskender, herewith declare that we have composed the present paper and worked ourselves and without use of any other than the cited sources and aids. Sentences or parts of sentences quoted literally are marked as such; other references with regard to the statement and scope are indicated by full details of the publications concerned. The paper and work in the same or similar form have not been submitted to any examination body and have not been published. This paper was not yet, even in part, used in another examination or as a course performance.