Tecnológico de Monterrey

Evidence 2. Review 3

Members

Eugenio Guzmán Vargas | A01621084

Diego Alberto Carrillo Castro | A01645484

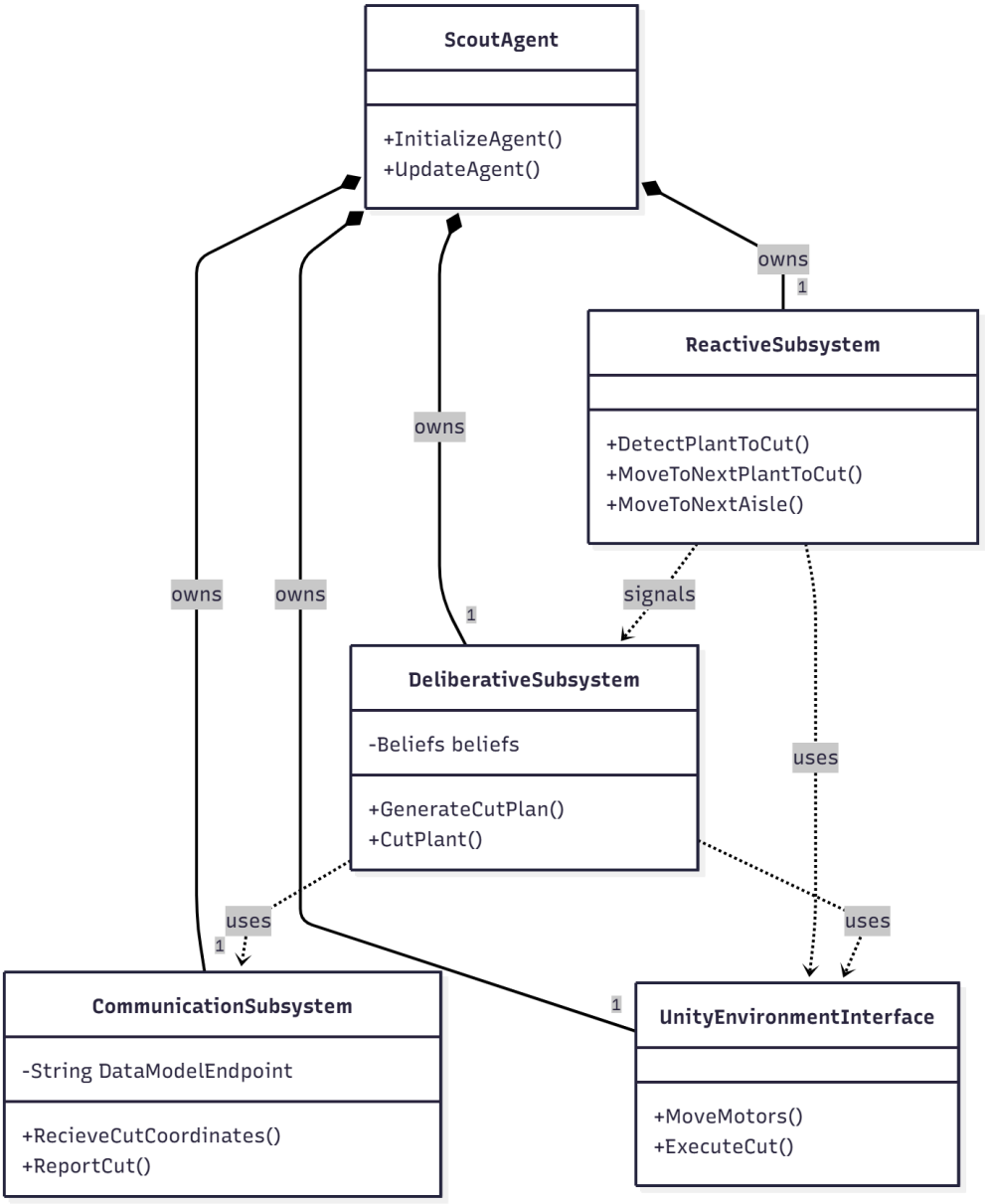
Braulio Fernando Antero Díaz | A01645356

Diego Núñez García | A01644371

Victor Manuel Laureano Vega | A01638979

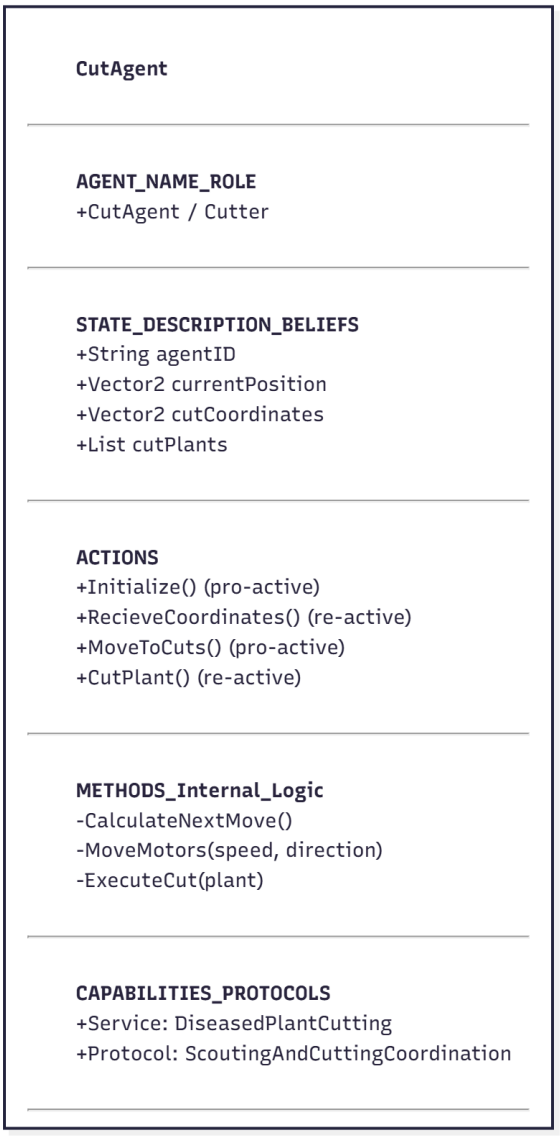
Modeling of Multi-Agent Systems with Computer Graphics

29/11/2025



This diagram represents the structure of the CutAgent, one of the two agents that will be present in our simulation. Its ReactiveSubsystem handles the logic for immediate responses, such as those required by DetectPlantToCut(), MoveToNextPlantToCut(), and MoveToNextAisle(), while the DeliberativeSubsystem is responsible for complex planning and decision-making, including GenerateCutPlan() and CutPlant().

The CommunicationSubsystem is responsible for establishing contact with the manager, which in turn establishes contact with the server where the neural network model is hosted. This model processes plant data and determines whether or not it is appropriate to cut the plant.



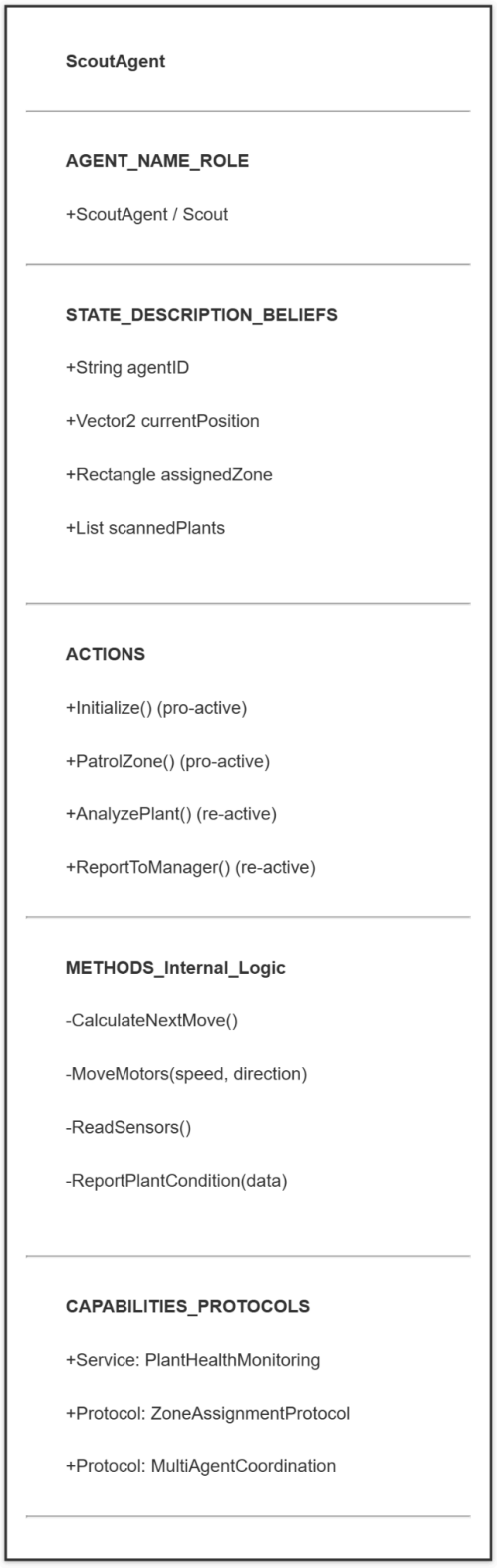
The AGENT_NAME_ROLE section identifies the class as CutAgent and its role as Cutter.

The STATE_DESCRIPTION_BELIEFS section contains the agent's working memory, including data for navigation (currentPosition), a list of the coordinates of the plants that must be cut (cutCoordinates), and a list of cutPlants, to keep control.

The ACTIONS section lists the agent's high-level capabilities. MoveToCuts is the main pro-active behavior, while CutPlant and RecieveCoordinates are reactive behaviors.

The METHODS_Internal_Logic section details the functions that implement the high-level ACTIONS. This includes MoveMotors for physical movement, CalculateNextMove to execute an intelligent movement in the environment, and ExecutCut.

The CAPABILITIES_PROTOCOLS section defines the agent's understanding of system-wide rules. ScoutingAndCuttingCoordination corresponds directly to a protocoled defined on the ScoutAgent, which helps them coordinate and, to some extent, communicate.



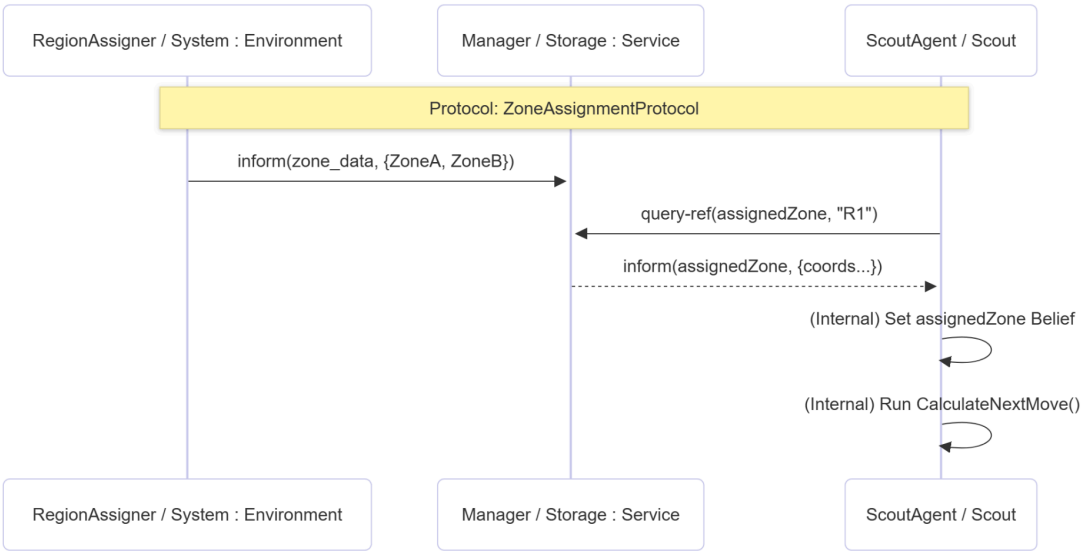
The AGENT_NAME_ROLE section identifies the class as ScoutAgent and its role as Scout.

The STATE_DESCRIPTION_BELIEFS section contains the agent's working memory. This includes key data for navigation (currentPosition), task execution (assignedZone), and a list of scannedPlants, to ensure work is not repeated.

The ACTIONS section lists the agent's high-level capabilities. PatrolZone is the main pro-active behavior, while AnalyzePlant and ReportToManager are reactive behaviors triggered by sensor events. ReportToManager is the agent's method of communicating its findings to the neural network model.

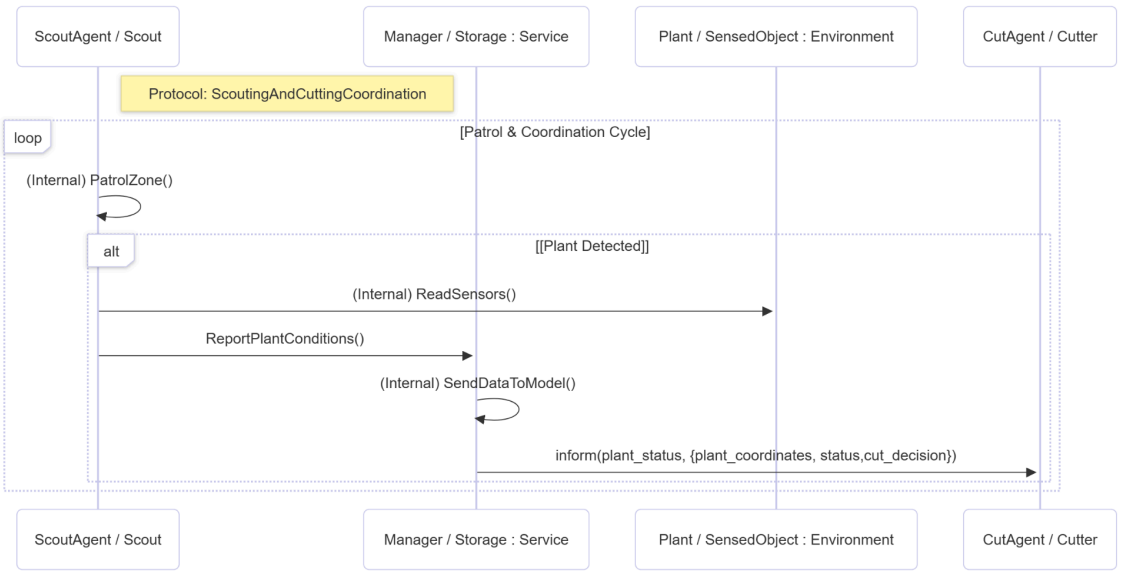
The METHODS_Internal_Logic section details the functions that implement the high-level ACTIONS. This includes MoveMotors for physical movement, CalculateNextMove to execute an intelligent movement in the environment, and ReportPlantCondition to handle data transmission.

The CAPABILITIES_PROTOCOLS section defines the agent's understanding of system-wide rules. ZoneAssignmentProtocol is used at startup to get its assigned area.



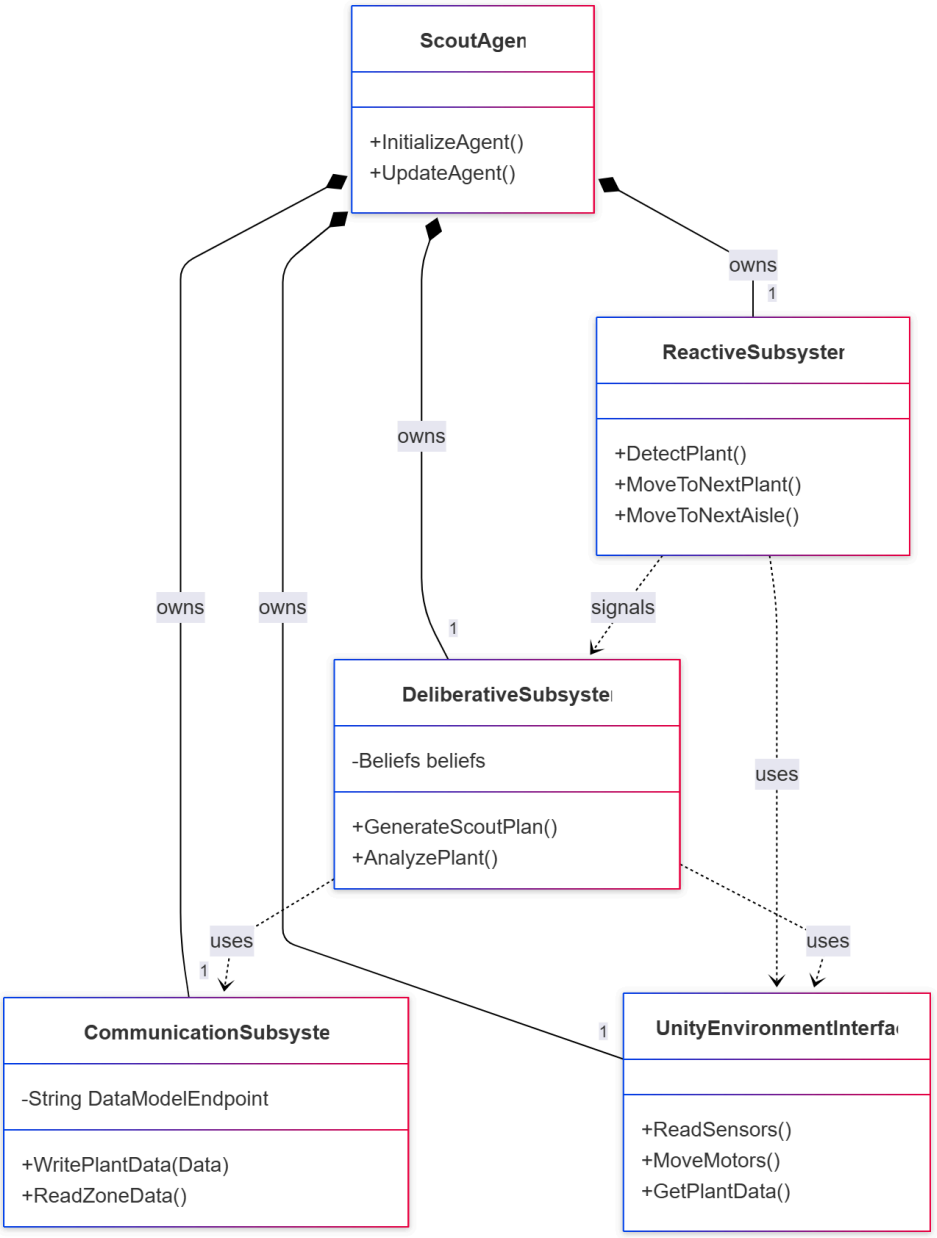
This diagram illustrates the ZoneAssignmentProtocol interaction.

When the ScoutAgent receives information about the zone they will be responsible for, they perform two internal steps: first, they update their belief with the assigned zone, and second, they execute a planning action to determine how to begin operating in their new work area.

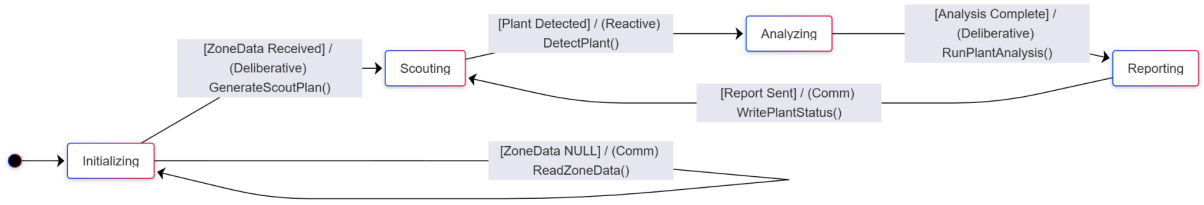


This diagram shows the ScoutingAndCuttingCoordination protocol. It visualizes the agent's main operational loop, driven by its PatrolZone() action. It also shows how it uses the manager to both report its status and read its peer's status, triggering the AvoidObstacle() action if necessary. It also shows the AnalyzePlant() and ReportToDatabase() sequences.

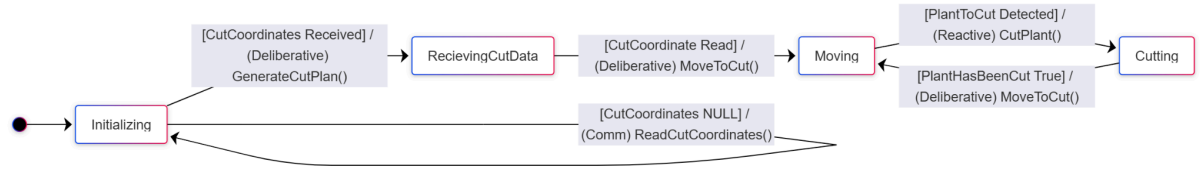
This diagram shows the Scouting and Cutting Coordination protocol. The ScoutAgent begins the main loop by patrolling its assigned zone. If a plant is detected anywhere along its path, the agent queries its data and reports it to the manager, who then internally queries the neural network model to evaluate whether cutting the plant is appropriate.



This diagram represents the structure of the ScoutAgent. Its ReactiveSubsystem handles the logic for immediate responses, such as those required by DetectPlant(), MoveToNextPlant (), and MoveToNextAisle(), while the DeliberativeSubsystem is responsible for complex planning and decision-making, including GenerateScoutPlan() and AnalyzePlant().



This diagram illustrates the ScoutAgent's behavior. From an initial state, if it has no zone data, it attempts to obtain it through the Communication subsystem. Once the data is received, it generates a ScoutPlan, moving to the Scouting state. From here, the Reactive subsystem detects a plant, transitioning to the Analyzing state. After completing the analysis, the Deliberative subsystem executes the analysis logic, and the agent moves to the Reporting state. Finally, after the Communication subsystem sends the plant's status, the agent returns to Scouting to continue its patrolling cycle.



This diagram illustrates the CutAgent's behavior. First, through the Communication subsystem, it attempts to acquire the cutting coordinates. Once received, the Deliberative subsystem generates a CutPlan, and the agent transitions to the ReceivingCutData state. From there, the agent moves to the Moving state, where the Deliberative subsystem directs it to the location of a plant that needs to be cut. Upon detecting the plant, the Reactive subsystem executes the cut, transitioning it to the Cutting state. Finally, after completing the cut, the Deliberative subsystem returns the agent to the Moving state to prepare for the next cutting task.

Work plan

The following is a breakdown of pending activities, the times in which they will be carried out and completed, as well as the team members who will be working on them.

Task	Responsible Team Member	Estimated effort	Estimated due date
Code the movement of the two ScoutAgent instances through the greenhouse in Unity.	Braulio, Diego N.	Low	Monday, 1st.
Code the movement of the CutAgent.	Braulio, Victor.	Low	Monday, 1st.
Enable the neural network to receive, in addition to the parameters necessary to evaluate the condition of the plants, their coordinates, in order to track them and relate them to the evaluation results.	Eugenio	Medium	Tuesday 2nd.
Code the ScoutAgent's internal functions related to locating and evaluating plants.	Braulio, Victor	Medium	Tuesday, 2nd.
Code the internal function that allows the ScoutAgent to communicate with the manager.	Diego C., Eugenio	High	Wednesday, 3th.
Code the CutAgent's internal functions related to locating and cutting plants.	Diego C., Diego N.	Medium	Tuesday, 2nd.

Code the CutAgent's internal function that allows it to receive information from the manager.	Eugenio, Victor.	High	Wednesday, 3th.
Perform tests of the complete simulation, from the ScoutAgent's movement, including the analysis of all plants, communication with the manager, and the cutting of diseased plants by the CutAgent.	All the team	Low	Thursday, 4th.