MAS Evidence 2

Sebastian Certuche A01644942

Contents

1 Multi-Agent Tomato Sorting System Final Report

1.1 Analysis of the Developed Solution

1.1.1 Why a Multi-Agent Model?

We selected a multi-agent system (MAS) instead of a single reactive agent or only BDI agents because:

Model Type	Limitation	Why It Does Not Fit
Pure reactive agents	No global planning, inefficient execution	Could get stuck or never finish mapping or collection
Pure BDI agents	High cognitive cost on each agent, vulnerable to conflicts	Harder to manage dynamic interactions like collision avoidance
Multi-agent architecture	Allows task specialization and distributed workload	Enables collaboration and adaptability

A MAS ensures efficient division of labor, allowing simultaneous mapping, inspecting, and collecting tasks. This reduces execution time and improves scalability.

1.1.2 Architecture Justification for Each Agent

Agent	Architecture Type	Why this architecture	Why not another architecture
Agent 1 (Mapping)	Reactive	Fast responses to the environment, no long-term reasoning needed after map creation	BDI would add unnecessary complexity without benefit

Agent	Architecture Type	Why this architecture	Why not another architecture
Agent 2 (Inspection)	BDI-like	Needs to classify tomatoes and decide when to report updated information	Pure reactive would not adapt to tomato state variability
Agent 3 (Collectors)	Reactive	Tasks are repetitive, fast execution improves harvesting efficiency	BDI would increase latency in collection cycles

Task specialization aligns each architecture with the operational constraints of its task.

1.1.3 Variables Considered in the Decision

Variable	Influence on the System
Number of plantations and tomatoes	Determines mapping complexity and task load
Obstacles and layout density	Affects navigation and collision probability
Tomato health states (healthy or infected)	Triggers classification and task routing
Agent distances and workloads	Used in dynamic task allocation
Real-time interactions	Requires communication to avoid conflicts

These factors impact performance, collision risk, and overall completion time.

1.1.4 Incidence Between Simulation Variables and Agent Architecture

Simulation Variable	Agent Most Affected	Architectural Impact on Results
Plantation quantity	Agent 1	Reactive design remains efficient under increased exploration demands
Health classification	Agent 2	Reasoning-based classification avoids incorrect routing
Collision risk	Agent 3	Reactive actions allow fast repositioning during congestion
Communication frequency	All agents	Coordination improves global efficiency and reduces deadlocks

Each architecture supports the variable that imposes the greatest constraint.

1.1.5 Justification of the Graphic Design

The environment features:

- A visible grid to emphasize spatial navigation
- Clear plantation objects representing agricultural obstacles
- Color-coded tomatoes and agents for rapid state recognition

This enhances debugging, spatial awareness, and comprehension of interactions.

1.1.6 Advantages of the Final Solution

- High modularity with isolated responsibilities
- Adaptive behavior driven by communication
- Efficient cost-based task allocation
- Reduced collisions through local negotiation

1.1.7 Disadvantages and Opportunities for Improvement

Disadvantage	Suggested Modification
Idle agents after completing tasks	Add patrol or predictive scanning behaviors
No continuous perception	Enable anomaly detection during movement
Congestion at deposits	Hybrid centralized and local path negotiation
Only two tomato categories	Expand classification (ripeness, disease type)
Static map	Support detection of dynamic obstacles

1.2 Learning Reflection

Initial expectations:

- Learn Unity for simulation building
- Understand MAS principles in an agricultural scenario
- Build a working prototype detecting infected tomatoes

Achievements:

- Designed agents with navigation, interaction, and cooperation
- Simulated realistic issues such as disease-driven waste in crops
- Applied methodologies for negotiation and conflict resolution
- Improved debugging, communication, teamwork, and scheduling

We strengthened both technical and organizational skills.

1.3 Final Description of the Multi-Agent System

1.3.1 Agent Roles

Agent 1 Maps the Area

- Detects plantations and obstacles
- Explores the grid and produces the global map

- Shares updated information with the system

Agent 2 Scans for Tomatoes

- Navigates plantations based on the map
- Inspects tomatoes and classifies them as healthy or infected

Agent 3 Cuts and Transports Tomatoes

- Harvests tomatoes and carries them to the correct deposit
- One collects healthy tomatoes and the other collects infected tomatoes
- Uses communication and navigation rules to avoid collisions

---

1.3.2 Communication Patterns

Pattern	Description
Activation messages	Agents confirm readiness before beginning operation
Heartbeat messages	Agents share their current location
Map updates	Global awareness is shared between agents
Task assignment	Proposals improve efficiency in workload distribution
Collision avoidance	Movement information prevents deadlocks and overlaps

---

1.4 Project Goals and Challenge Summary

Goals:

- Implement autonomous mapping, classification, and harvesting
- Optimize actions through coordination and communication
- Simulate realistic tomato farming logistics

Real impact: tomato production faces major losses without automation. Intelligent agents reduce waste and improve sustainability.

1.5 Conclusion

The implemented MAS successfully achieves:

- Autonomous warehouse exploration
- Tomato recognition and classification
- Efficient cooperative harvesting
- Collision-free navigation
- Realistic agricultural modeling

This forms a scalable foundation for future intelligent farming systems.