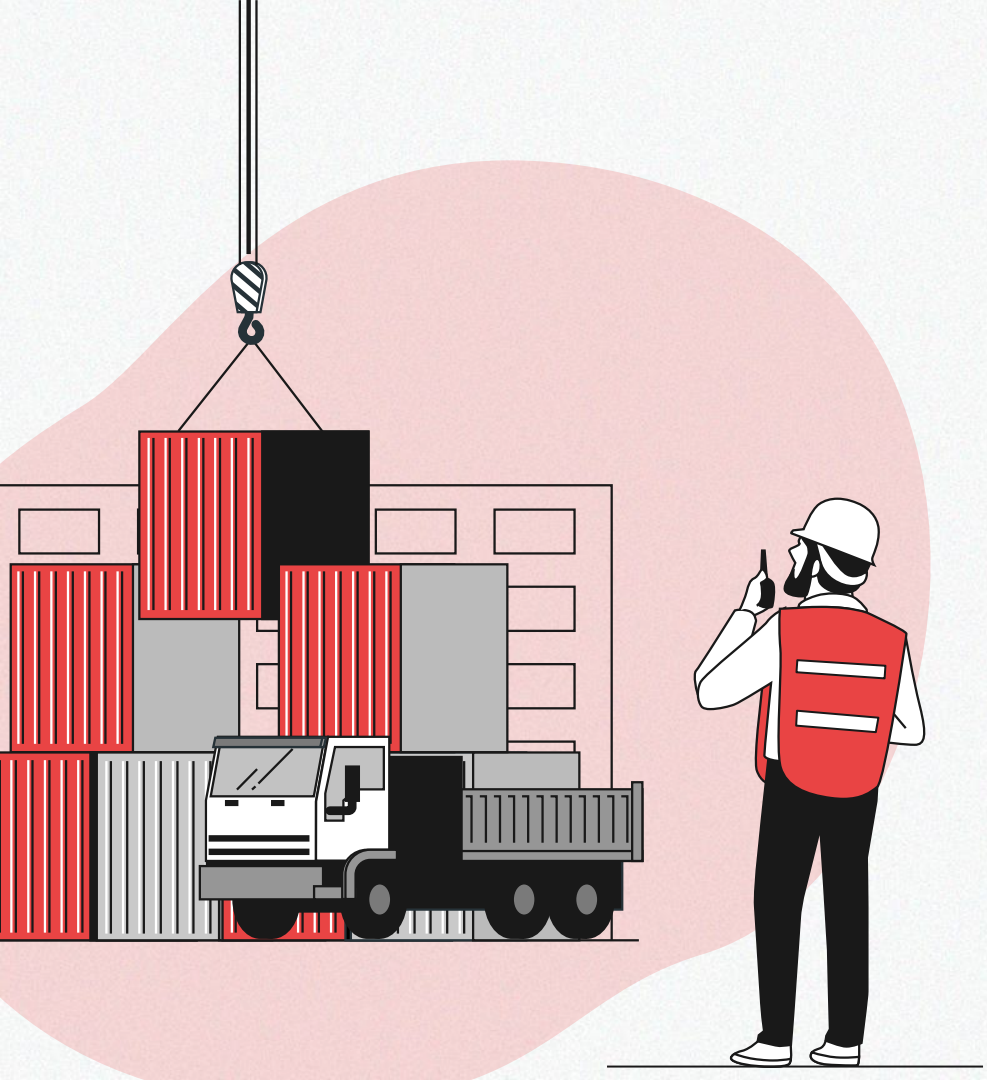


INTELLIGENT WAREHOUSE SYSTEM

Roberto Di Via 4486648

Università di Genova

TABLE OF CONTENTS

01.

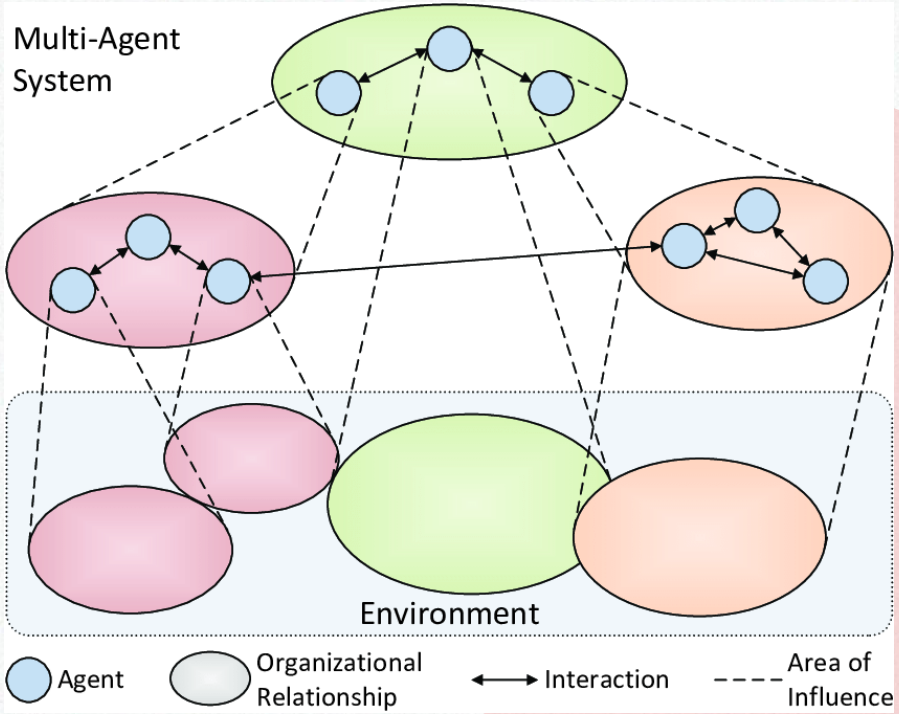
INTRODUCTION

MULTI-AGENTS SYSTEM

A multi-agent system (MAS) is a group of agents that interact with each other and the environment to achieve goals.

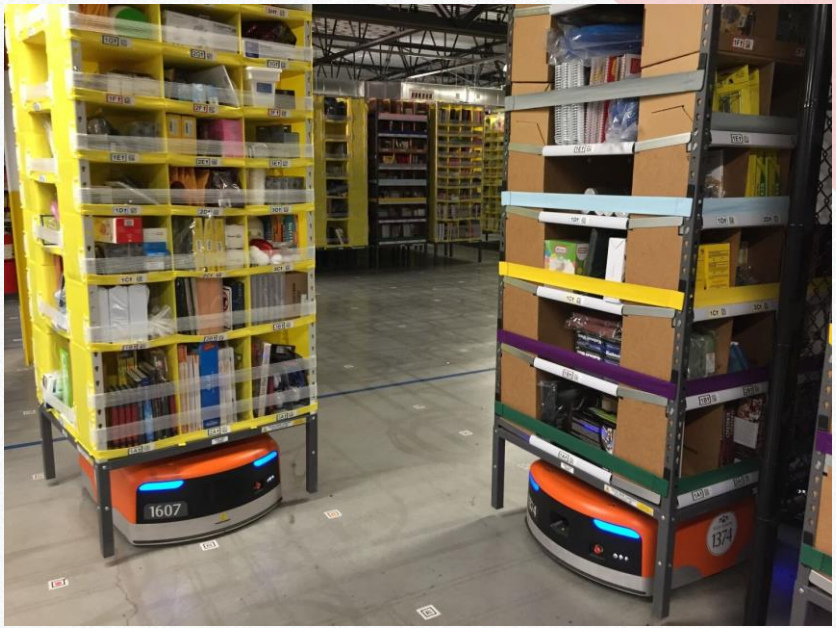
Each agent in the system is autonomous and has its own local view, meaning that no agent has a complete global view of the system.

MAS can solve complex problems that are difficult or impossible for an individual agent or a monolithic system to solve.

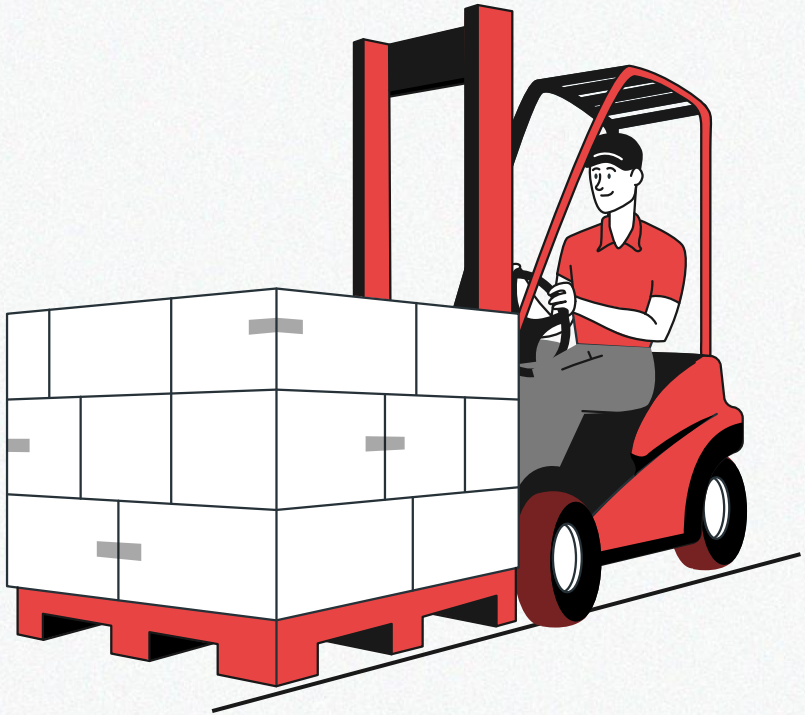


WHY A WAREHOUSE?

- Complex tasks requiring coordination among multiple agents
- Improving warehouse flexibility and adaptability
- Better fault tolerance and robustness in warehouse operations



02.
TECHNOLOGIES USED



JAVA

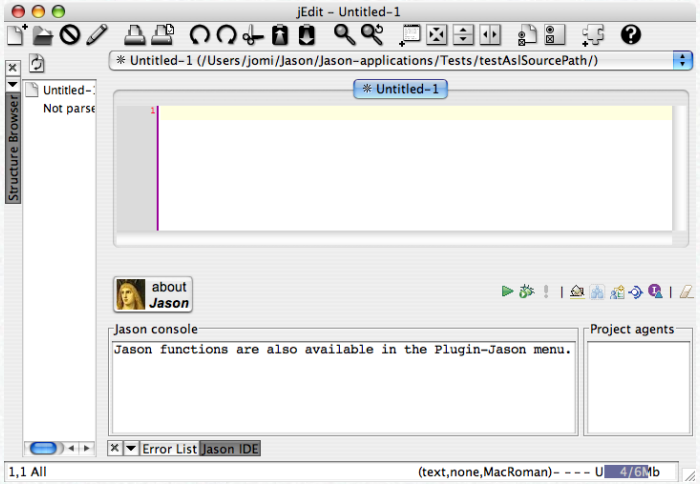
Object-Oriented programming language

- Define environment's state
- Provides a Graphical User Interface (GUI)
- Handle environmental percepts
- Execute agents' actions

JASON

Open-source interpreter for AgentSpeak

- Provides a platform for MAS development
- Enhances AgentSpeak with additional features and constructs
- Program agents' behaviors and decision-making



AGENTSPEAK(L)

Logic-based agent-oriented programming language

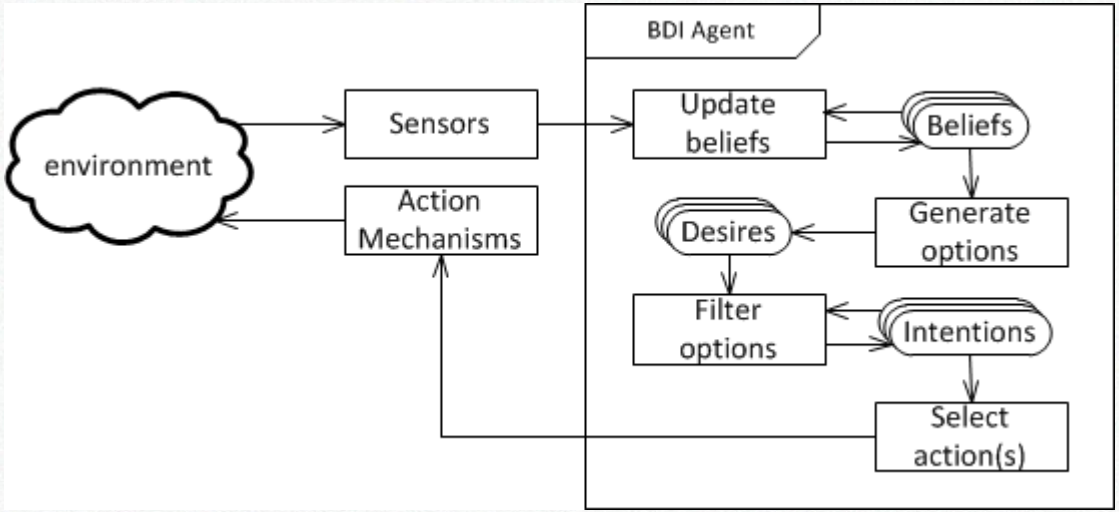
- Based on BDI agent architecture
- Enables agents to reason about beliefs, desires, and intentions
- Supports decision-making and action execution based on reasoning

$$
\begin{array}{lll}
ag & ::= & bs \quad ps \\
bs & ::= & at_1 \ldots at_n \qquad (n \geq 0) \\
at & ::= & P(t_1, \ldots t_n) \qquad (n \geq 0) \\
ps & ::= & p_1 \ldots p_n \qquad (n \geq 1) \\
p & ::= & te : ct \leftarrow h \\
te & ::= & +at \mid -at \mid +g \mid -g \\
ct & ::= & at \mid \neg at \mid ct \ \wedge \ ct \mid \mathsf{T} \\
h & ::= & a \mid g \mid u \mid h; h \\
g & ::= & !at \mid \ ?at \\
u & ::= & +at \mid -at
\end{array}
$$

BDI ARCHITECTURE

Belief-Desire-Intention architecture

- Beliefs: Agent's knowledge about the world
- Desires: Agent's goals
- Intentions: Agent's plans for achieving goals



03.
SYSTEM ARCHITECTURE

MAS MAIN'S COMPONENTS

ENVIRONMENT

The space where agents interact and perform actions

AGENTS

Autonomous entities with specific goals and decision-making abilities

COMMUNICATION

Enables agents to exchange information and coordinate actions

ENVIRONMENT

15x15 grid cells

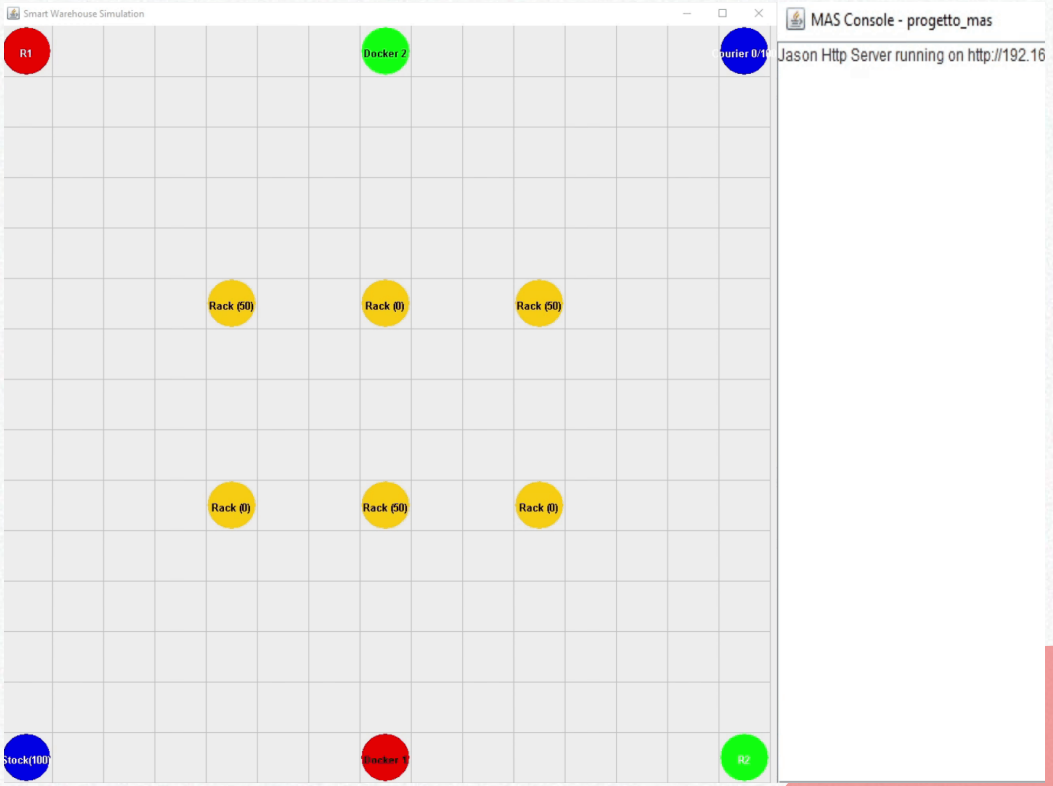
Basic path search algorithm for agent movement

Percepts:

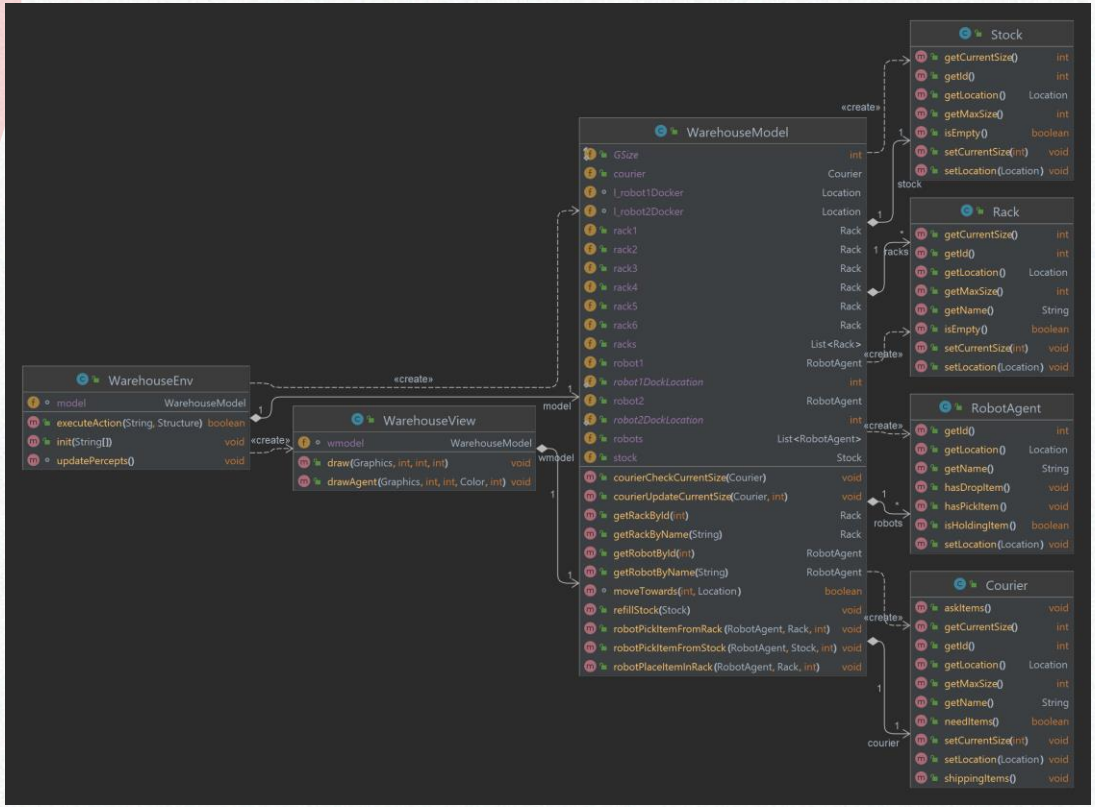
- object and agent locations
- stock availability
- rack state

COMPONENTS

- 6 racks
- Stock
- Courier
- 2 Robots
- 2 Docker stations

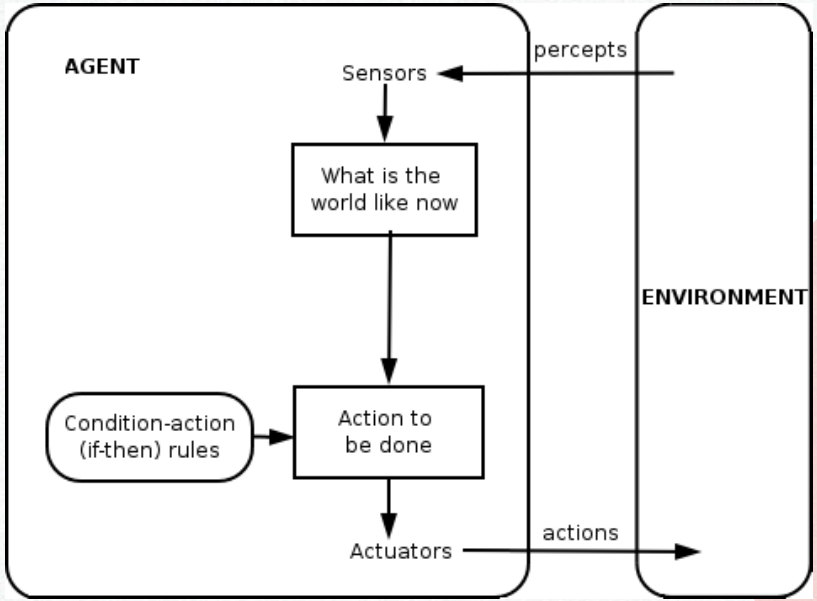


JAVA COMPONENTS AND RELATIONSHIPS



AGENTS DEFINITION

Agents are autonomous software entities that perceive their environment through sensors and make decisions based on their internal rules or conditions. They then perform actions or interact with the environment through actuators to achieve their goals.



AGENTS AND THEIR ROLES

ROBOT 1

- Fills racks with items
- Picks items from Stock agent

ROBOT 2

- Handles Courier agent's requests
- Picks items from filled racks and delivers them to Courier

STOCK

- Manages item availability
- Refills items when empty (requested by Robot 1)

COURIER

- Handles deliveries to buyers
- Requests items from Robot 2 every 10 seconds

AGENTS INTERACTIONS AND COMMUNICATION

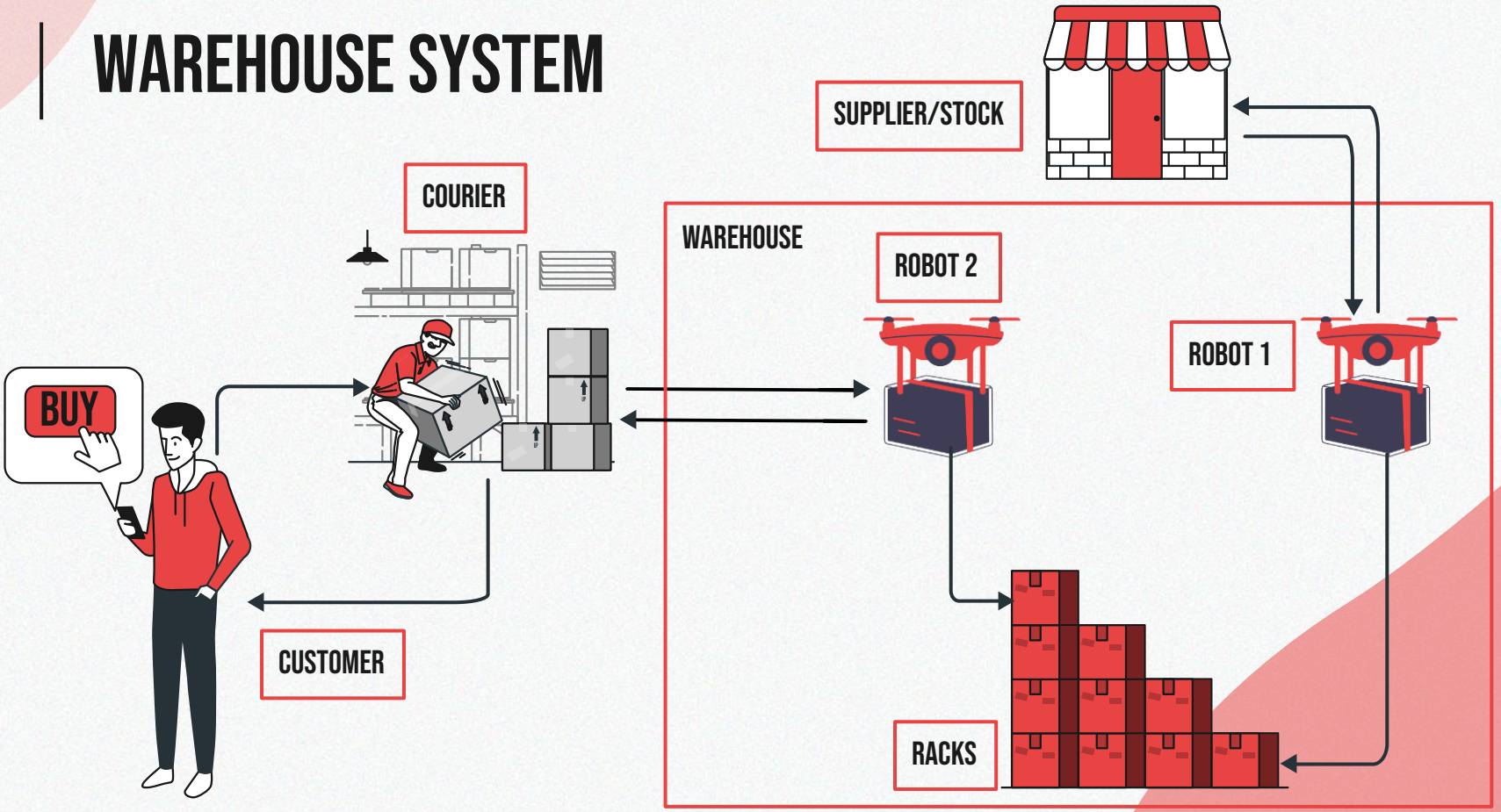
AGENT INTERACTIONS

- Robot 1 and 2 perceive rack states from the environment.
- Robot 1 interacts with Stock to ask the refill and waits for the event.
- Courier interacts with Robot 2 to requests items.
- Robot 2 interacts with the Courier to inform him about the items delivered.

COLLABORATION BENEFITS

- Efficient warehouse operations
- Stock never empty
- Racks always filled
- Courier requests satisfied

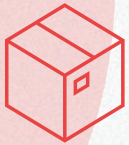
WAREHOUSE SYSTEM



04.

CONCLUSIONS

FUTURE WORKS



DYNAMIC REQUEST QUANTITY

Enhance courier agent's capability to request variable item quantities.

This can improves flexibility and responsiveness.

ADVANCED PATHFINDING ALGORITHMS

Enhance navigation efficiency to prevent overlaps and minimize travel distances.

This can optimize resource utilization and reduce delivery times.

GOAL SUSPENSION AND TASK REASSIGNMENT

Mechanisms for goal suspension and reassignment during recharging periods.

This can ensure robustness and a better fault tolerance.

“Individually, we are one drop.
Together, we are an ocean.”

— RYUNOSUKE SATORO

THANKS!

Do you have any questions?

RESOURCES

- AI for Anyone. Multi-agent system. url: https://www.aiforanyone.org/glossary/multi-agent-system.
- Universit`a di Bologna. AgentSpeak(L) and Jason. url: https://core. ac.uk/download/pdf/17198968.pdf.
- Jason. Java-based interpreter. url: https://jason.sourceforge.net/ wp/.
- Oracle. Java Technical Details. url: https://www.oracle.com/java/ technologies/.
- Michael Wooldridge Rafael H. Bordini Jomi Fred Hubner. Programming Multi-Agent Systems in AgentSpeak using Jason. 2007.
- Wikipedia. Belief-Desider-Intention architecture. url: https://en.wikipedia.org/wiki/Belief%E2%80%93desire%E2%80%93intention_software_model
- Wikipedia. Unified Modeling Language. url: https://en.wikipedia.org/wiki/Unified_Modeling_Language.