Università degli Studi dell'Aquila

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Student Name: Dima Mhrez

Student ID: 301660

Supervisor: Stefania Costantini

Supervisor: Giovanni De Gasperis

1 Introduction

This project presents a **Multi-Agent System (MAS) for smoke emergency response** implemented in DALI Prolog. The system simulates how intelligent agents cooperate to detect, manage, and respond to smoke incidents inside a building.

In real-world environments such as hospitals, offices, and public facilities, early smoke detection and rapid coordination are essential to ensure people's safety and reduce infrastructure damage. This project models a simplified but realistic emergency scenario in which specialized agents collaborate to handle smoke alerts and evacuation procedures.

The main objective is to show how multiple agents can work together to provide a structured and intelligent emergency response.

1.1 System Overview

The Emergency Smoke Response MAS consists of four main agents that interact to detect and manage smoke emergencies:

1. Room Sensor Agent
2. Safety Manager Agent

3. Guard Team Agent
4. Evacuator Agent

Each agent has a specific responsibility and communicates with others using DALI message passing.

When smoke exceeds a safety threshold, the system triggers a coordinated response involving inspection and evacuation.

1.1.1. Roles

The Emergency Smoke Response Multi-Agent System is composed of specialized agents, each with a clearly defined role. The separation of responsibilities allows efficient coordination and fast reaction to emergency situations.

Role	Description / Main Responsibilities
Room Sensor Agent	Monitors smoke levels in rooms. When the measured value exceeds a predefined threshold, it generates an alarm and notifies the Safety Manager.
Safety Manager Agent	Acts as the central coordinator. Receives alarms from sensors, manages multiple pending alarms, and dispatches the Guard Team and Evacuator. Ensures that emergencies are handled sequentially without conflict.
Guard Team Agent	Responds to the physical location of the emergency. Verifies the smoke level and, if the situation is critical, escalates the emergency by calling firefighters**.**
Evacuator Agent	Manages evacuation procedures. Unlocks doors, activates warning signals (sirens), and ensures that occupants can safely leave the affected area. Sends confirmation once evacuation is complete.

1.1.2 Role Schemas

Role Schema: RoomSensor

Description

Monitors smoke levels in each room.
When a new smoke level new_smoke(Level,Room) is received from the environment, the agent checks the configured threshold.
If Level ≥ threshold, it raises an alarm and informs the SafetyManager by sending alarm(Level,Room).

Protocols and Activities

External event: new_smoke(Level,Room) (from environment)

Internal state trigger: smoke_received(Level,Room) → smoke_receivedI/2

Action/message: send alarm(Level,Room) to SafetyManager

Permissions

Reads

threshold**(Th)**

Writes

smoke_received(Level,Room) (temporary internal state)

Generates (messages)

alarm(Level,Room) → SafetyManager

Responsibilities

Liveness

new_smoke(L,R) ∧ L ≥ Th → alarm(L,R)

Safety

Alarms are generated only when the threshold is exceeded (L ≥ threshold).

Role Schema: SafetyManager

Description

Acts as the coordinator of the smoke emergency response.
Receives alarm(Level,Room) messages from the RoomSensor, stores alarms in a local queue (pending_alarm/2), and dispatches the GuardTeam and Evacuator.
Uses a busy/0 flag to ensure that only one emergency is handled at a time.

Protocols and Activities

External events: alarm(Level,Room), guard_confirm(Room), firefighters_called(Room), room_safe(Room)

Internal events: pending_alarmI(Level,Room), trigger_next_dispatch

Actions/messages: dispatch_guard(Room,Level), start_evacuation(Room,Level)

Permissions

Reads (state)

pending_alarm(Level,Room)

busy/0

Writes (state)

pending_alarm(Level,Room)

busy/0

Generates (messages)

dispatch_guard(Room,Level) → to GuardTeam

start_evacuation(Room,Level) → to Evacuator

Responsibilities

Liveness

alarm(L,R) → pending_alarm(L,R)

If \+ busy then: pending_alarm(L,R) → dispatch_guard(R,L) ∧ start_evacuation(R,L)

room_safe(R) → trigger_next_dispatch

Safety

Handles **at most one** emergency at a time (busy/0 prevents parallel dispatch).

Additional alarms are **queued** as pending_alarm/2 and processed sequentially.

Role Schema: GuardTeam

Description

Responds to the emergency location and evaluates severity.
Calls firefighters if smoke levels are critical.

Protocols and Activities

dispatch_guard, guard_confirm, firefighters_called

Permissions

Reads

smoke_level(Level)

Generates

guard_confirm(Room)

firefighters_called(Room)

Responsibilities

Liveness

dispatch_guard(R,L) → guard_confirm(R)

Safety

Calls firefighters only in critical situations

Role Schema: Evacuator

Description

Executes evacuation procedures in the affected room upon request from the SafetyManager. Performs evacuation actions (unlock doors, activate sirens) and confirms completion by notifying the SafetyManager with room_safe(Room).

Protocols and Activities

External events: start_evacuation(Room,Level)

Internal events: evac_requestI(Room,Level), evac_doneI(Room)

Outgoing message: room_safe(Room) (to SafetyManager)

Permissions

Reads (state)

evac_request(Room,Level)

evac_done(Room)

Writes (state)

evac_request(Room,Level)

evac_done(Room)

Generates (messages)

room_safe(Room) → to SafetyManager

Responsibilities

Liveness

start_evacuation(R,L) → evac_request(R,L) → evac_done(R) → room_safe(R)

Safety

Declares the room safe only after executing evacuation actions (unlocking doors and activating the siren).

Ensures the internal request is consumed (retract/1) to avoid repeated evacuation confirmations.

1.2 Virtual Organization

Name: EmergencySmokeResponse

Goals:

- Minimize risk to people and infrastructure.
- Ensure prompt and coordinated reaction to smoke emergencies.

- Support distributed decision-making among agents.
- Guarantee safe evacuation procedures.

Roles and Interactions:

- **RoomSensor → SafetyManager:** sends alarm messages when smoke exceeds the threshold.
- **SafetyManager → GuardTeam:** dispatches the guard team to inspect the affected room.
- **SafetyManager → Evacuator:** sends evacuation commands for the affected location.
- **GuardTeam → SafetyManager:** confirms arrival and reports if firefighters are needed.
- **Evacuator → SafetyManager:** informs when evacuation and safety procedures are completed.

1.3 Event Table

RoomSensor

Event	Type	Source
new_smoke(L,R)	external	environment
smoke_received(L,R)	internal	state
fire_alarm(L,R)	internal	state

SafetyManager

Event	Type	Source
alarm(L,R)	external	RoomSensor
guard_confirm(R)	external	GuardTeam
firefighters_called(R)	external	GuardTeam
room_safe(R)	external	Evacuator
pending_alarm(L,R)	internal	state
trigger_next_dispatch	internal	state

GuardTeam

Event	Type	Source
dispatch_guard(R,L)	external	SafetyManager

Evacuator

Event	Type	Source
start_evacuation(R,L)	external	SafetyManager
evac_request(R,L)	internal	state
evac_done(R)	internal	state
room_safe(R)	external	Evacuator

1.4 Action Table

RoomSensor

Action	Description
alarm(L,R)	Sends smoke alarm to SafetyManager

SafetyManager

Action	Description
dispatch_guard(R,L)	Sends guard team to inspect room
start_evacuation(R,L)	Sends evacuation command to Evacuator
queue_alarm	Stores alarm when busy

GuardTeam

Action	Description
guard_confirm(R)	Confirms arrival to SafetyManager
firefighters_called(R)	Requests firefighters for severe cases

Evacuator

Action	Description
evacuate_room	Starts evacuation procedure
unlock_doors	Opens doors for safe exit
siren_on	Activates warning signals
room_safe(R)	Reports evacuation completed

1.5 Agent Behaviors

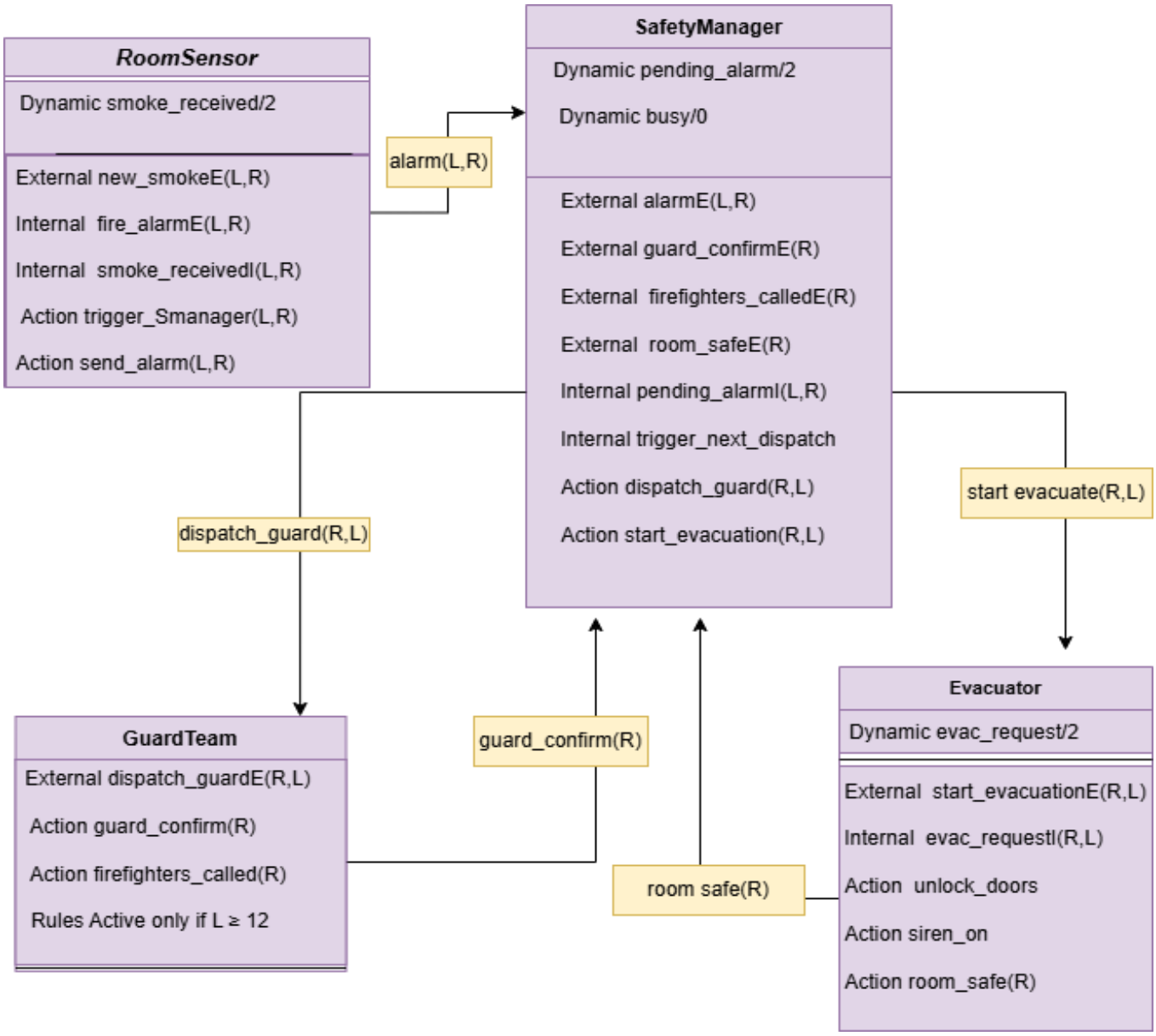
- **RoomSensor**:
  Proactive; continuously monitors smoke levels and generates alarms when the threshold is exceeded. Filters normal readings from dangerous ones and triggers the emergency workflow. Proactive in its use of internal states and events.

- **SafetyManager**:
  Reactive to incoming alarms; proactive in coordinating the emergency response. Manages multiple alarms using internal states (busy and pending alarms) to ensure that emergencies are handled sequentially. Dispatches the GuardTeam and Evacuator and monitors completion of tasks.

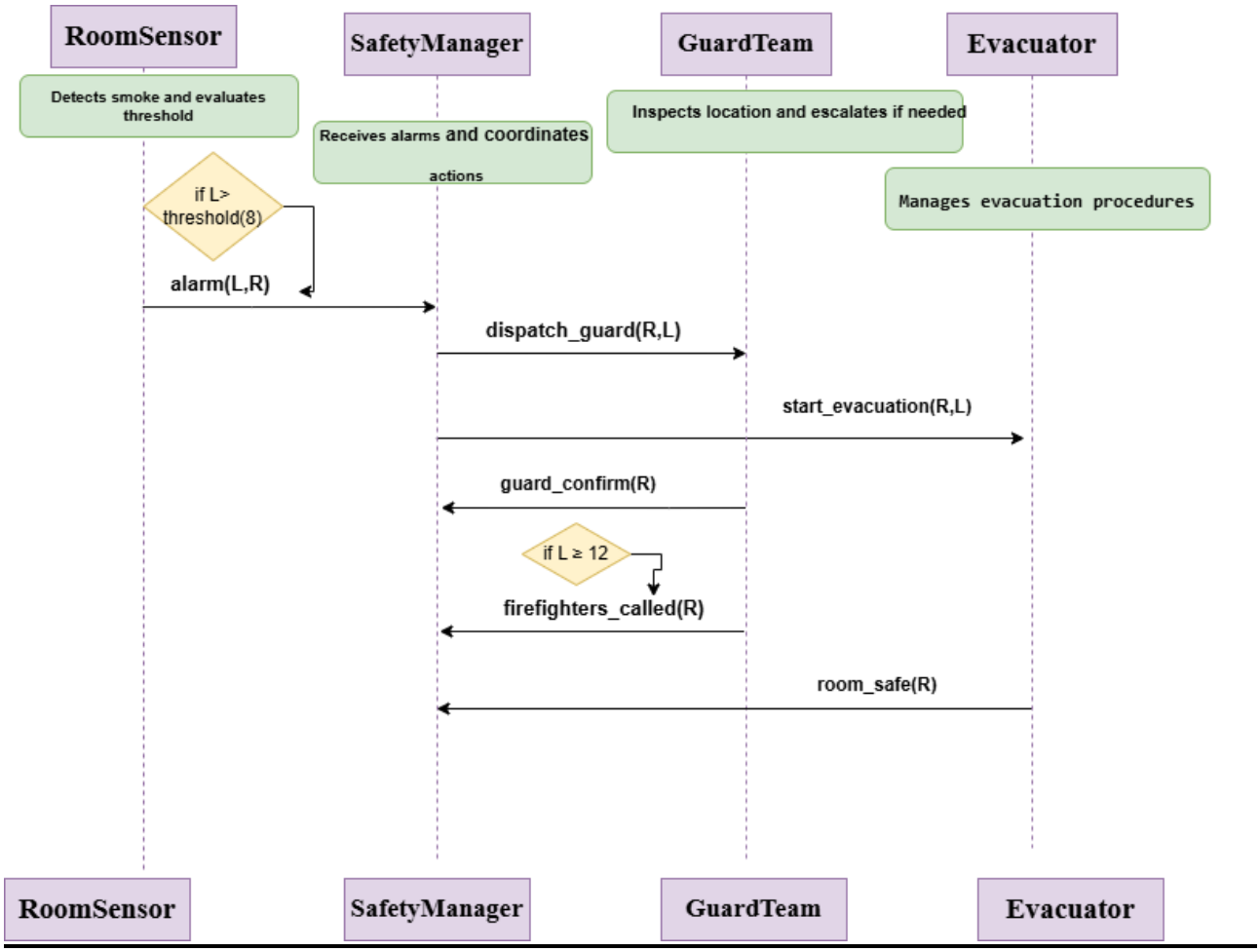
- **GuardTeam**:
  Reactive to dispatch commands. Inspects the affected location and confirms arrival. Proactive in escalation when smoke levels are critical by requesting firefighter intervention.

- **Evacuator**:
  Reactive to evacuation commands. Executes evacuation procedures such as unlocking doors and activating sirens. Reports confirmation when the room is safe. Uses internal states to manage evacuation steps.

Class Diagram - Smoke emergency response

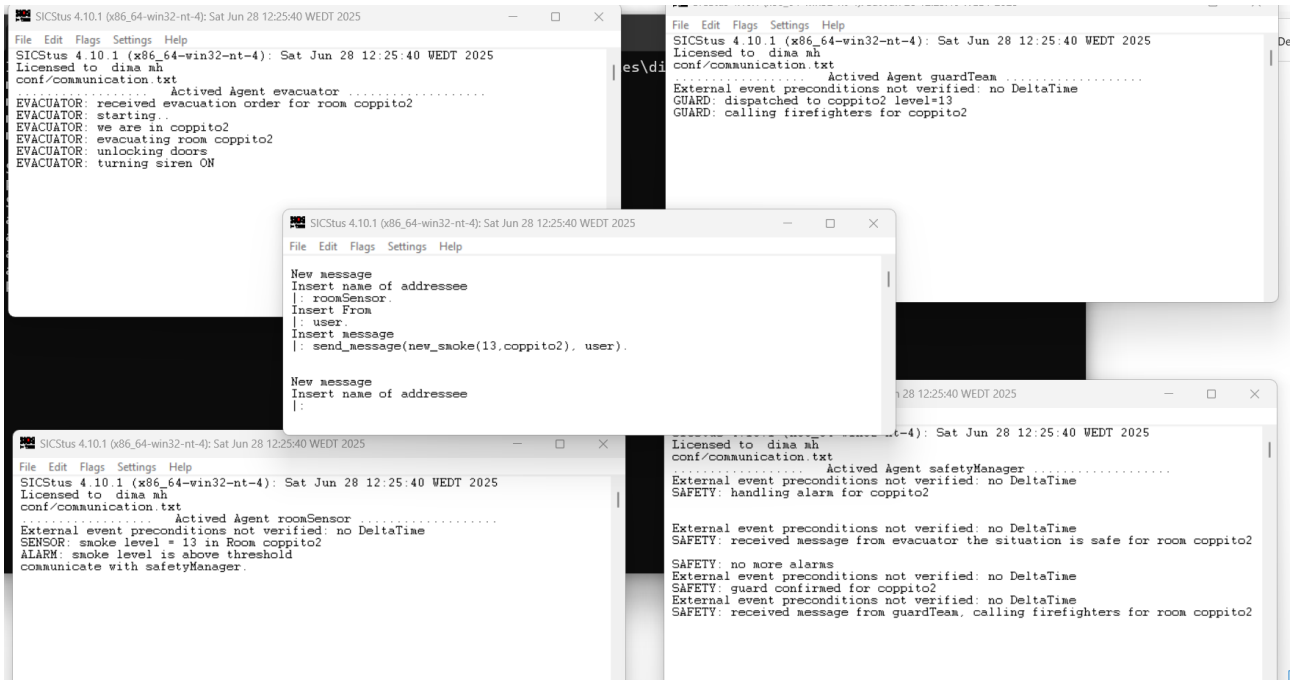


Sequence Diagram :



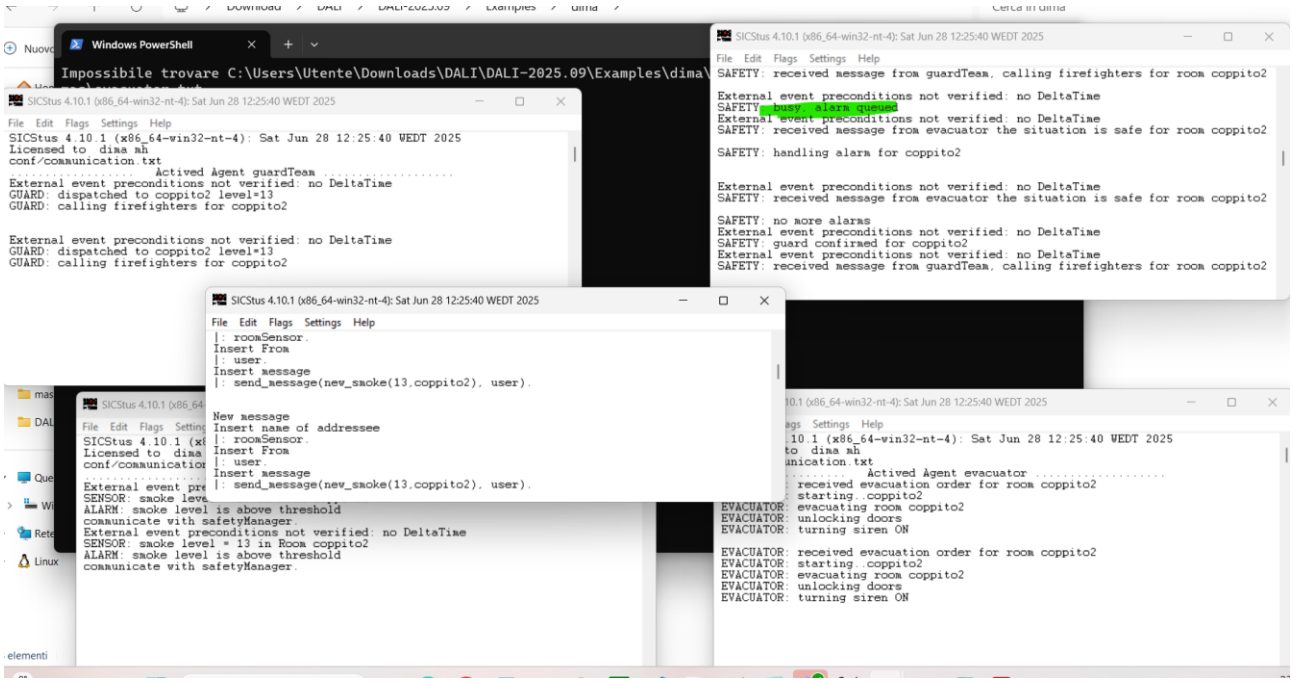
1.6 System Execution and Validation

This section documents the execution of the Emergency Smoke Response MAS and demonstrates that agents interact correctly during a smoke emergency.



To validate the queue mechanism implemented in the SafetyManager, we manually injected two smoke detections from the DALI “New message” console.
The message was sent to the roomSensor agent using:

- send_message(new_smoke(Level, Room), user).



1.7 System Execution and Results

This section describes the runtime behavior of the Emergency Smoke Response MAS and explains the outputs produced by each agent during execution.

1.7.1 Observed Agent Behavior

RoomSensor Output

- Detected smoke level in the room
- Verified that the threshold was exceeded
- Sent alarm to SafetyManager

This confirms the sensor correctly filters dangerous situations.

SafetyManager Output

- Received the alarm
- Handled the emergency request
- Coordinated evacuation and guard dispatch
- Confirmed when the room became safe
- • Set **`busy`** state while handling the current alarm (prevents parallel handling).
- • Queued additional alarms as `pending_alarm(Level,Room)` when another alarm arrives during busy.
- • Released **`busy`** only after **`room_safe(Room)`**
-

This shows proper coordination and alarm handling.

GuardTeam Output

- Dispatched to the affected room
- Verified smoke level
- Confirmed presence to SafetyManager
- Since the smoke level is below the critical threshold (12), firefighters are not required.

This demonstrates correct response to dispatch commands.

Evacuator Output

- Received evacuation command
- Unlocked doors
- Activated sirens
- Reported evacuation completion

This confirms evacuation procedures are executed.

1.7.2 System Validation

The execution logs demonstrate that:

- Agents communicate correctly via message passing
- The threshold mechanism works properly
- Emergency handling is coordinated
- Evacuation procedures are triggered

Repository

https://github.com/dimmhr/DALI-Smoke-Emergency-Response.git