Implement a Multiagent System for Campus Virtual Tour

- Tushar Sharma

Objective:

This assignment aims to design and implement a multiagent system to facilitate visitors navigating a campus and meeting their hosts. The system will include communication protocols between Campus Incharge (CI) agents and Building Incharge (BI) agents, utilizing the CrewAI framework and ROS for agent implementation and simulation.

System Description:

1. Agents and Roles:
    - Campus Incharge Agents (CI Agents):
        - CI agents are responsible for navigating (escorting) visitors from the campus entrance to the building where the host is located.
        - CI agents have access to the campus map but do not have access to individual building maps. They need to communicate with BI agents to navigate within a building.
    - Building Incharge Agents (BI Agents):
        - Each building has exactly one BI agent responsible for the navigation map of the respective building.
        - BI agents provide navigation assistance to CI agents, disclosing paths from the building entrance to the host's location based on the visitor's authorization.
        - BI agents can deny access if the visitor lacks authorization or if the host is unavailable in the building at that time.
    - Visitor Agents:
        - Visitor agents represent people visiting the campus to meet a host.
        - Visitors are guided by CI agents from the campus entrance to the host's location.
2. Agent Interactions:
    - CI agents meet visitors at the campus entrance and escort them to the relevant building.
    - Inside a building, the CI agent communicates with the BI agent to get the navigation path to the host's location.
    - If it so happens (as a special case) that a visitor wants to meet the BI agent itself (i.e. the BI agent will need to serve as host), the BI agent (while serving as host to the visitor) will go out of service (OOS) and will not entertain new navigation requests for the specified time.

- The OOS response will be accompanied by an integer stating the time duration for which the BI agent is out of service. In case the BI agent exceeds the committed time to be out of service, a violation event will be generated.

3. Performance Metrics:
   - CI Agents: Performance is measured based on the number of visitors entertained and penalties for violation events (e.g., not returning on time to pick up visitors).
   - BI Agents: Performance is based on the number of CI agents guided and penalties for violation events (e.g., failing to adhere to the OOS duration).

Software Design Report: Campus Tour Simulation

1. ROS2-based Implementation (without CrewAI)

System Architecture

The ROS2-based implementation follows a distributed architecture with multiple nodes:

1. Campus Incharge (CI) Agent Node
2. Building Incharge (BI) Agent Nodes (one per building)
3. Visitor Agent Nodes
4. Main Simulation Node (optional, for coordinating the simulation)

Each node is implemented as a ROS2 node, allowing for distributed processing and easy scaling.

Agent Communication Protocols

Communication between agents is achieved using ROS2 topics:

- /ci_requests: For requests to the CI agent
- /ci_responses: For responses from the CI agent
- /bi_requests: For requests to BI agents
- /bi_responses: For responses from BI agents
- /visitor_status: For publishing visitor status updates

Messages are serialized as JSON strings within ROS2 String messages, allowing for flexible message structures.
I set 10 time bounds for the ci_agent.

Simulation Setup and Scenarios

The simulation is set up using a ROS2 launch file, which starts multiple instances of each node type:

- One CI agent node
- Multiple BI agent nodes (one per building)
- Multiple visitor agent nodes

Scenarios include:
1. Visitors requesting campus navigation from CI agents
2. Visitors requesting building navigation from BI agents
3. Handling busy CI agents and retry mechanisms
4. BI agents going out of service for meetings

Performance Analysis

Pros:
- Distributed architecture allows for easy scaling and potential real-world deployment
- ROS2's publish-subscribe model enables loose coupling between components
- Can leverage ROS2's built-in tools for debugging and visualization

Cons:
- Increased complexity due to distributed nature
- Potential overhead from inter-process communication
- Limited to pre-programmed behaviors without advanced decision-making capabilities

2. CrewAI-based Implementation

System Architecture

The CrewAI-based implementation uses a centralized architecture with autonomous agents:

1. CIAgent class (subclass of CrewAI's Agent)
2. BIAgent class (subclass of CrewAI's Agent)
3. CampusSimulation class (orchestrates the simulation)

All components run within a single Python process, with CrewAI managing agent interactions.

Agent Communication Protocols

Communication between agents is handled internally by CrewAI:

- Agents are defined with roles, goals, and backstories
- Tasks are created and assigned to agents
- CrewAI's Crew class manages task execution and agent interactions

There are no explicit message passing or protocols; CrewAI handles inter-agent communication implicitly.

Simulation Setup and Scenarios

The simulation is set up in the CampusSimulation class:

- One CI agent is created
- Multiple BI agents are created (one per building)
- Visitors are simulated as tasks assigned to agents

Scenarios include:

1. CI agent escorting visitors to buildings
2. BI agents guiding visitors within buildings
3. Agents providing information about the campus and buildings

Performance Analysis

Pros:

- Simplified setup and execution (single Python script)
- Leverages AI for more dynamic and adaptable agent behaviors
- Easy to modify and extend agent capabilities

Cons:

- Runs in a single process, potentially limiting scalability for large simulations
- Dependent on external AI services, which may introduce latency or costs
- Less suitable for real-time or distributed robotic systems compared to ROS2

Comparison and Interpretation of Results

1. Flexibility:
   - ROS2: More flexible for real-world robotics applications
   - CrewAI: More flexible for simulating complex decision-making

2. Scalability:
   - ROS2: Better for large-scale, distributed systems
   - CrewAI: Better for increasing complexity of agent behaviors

3. Realism:
   - ROS2: More realistic for simulating actual robotic systems
   - CrewAI: More realistic for simulating human-like decision-making

4. Development Complexity:
   - ROS2: Higher initial setup complexity, but leverages existing robotics ecosystem
   - CrewAI: Lower initial setup complexity, but may require more effort for complex scenarios

5. Performance:
   - ROS2: Potentially faster execution, especially for simple behaviors
   - CrewAI: May have higher latency due to AI processing, but capable of more complex behaviors

In conclusion, the choice between these implementations depends on the specific goals of the simulation. The ROS2-based system is more suitable for simulating realistic robotic interactions and distributed systems, while the CrewAI-based system is better for exploring complex, adaptive agent behaviors in a simplified environment.

Simulation Visualizations and Logs:

- Graphical visualizations of the simulation environment.
- Communication logs between agents during the simulation.

Simulation-

```
tushar45@LAPTOP-OQEBKQF1:~/ros2_ws/src/campus_virtual_tour$ colcon build --packages-select campus_virtual_tour

Starting >>> campus_virtual_tour
Finished <<< campus_virtual_tour [2.28s]

Summary: 1 package finished [3.66s]
tushar45@LAPTOP-OQEBKQF1:~/ros2_ws/src/campus_virtual_tour$
tushar45@LAPTOP-OQEBKQF1:~/ros2_ws/src/campus_virtual_tour$ source install/setup.bashtushar45@LAPTOP-OQEBKQF1:~/ros2_ws/src/campus_virtual_tour$ ros2 launch campus_virtual_
tour campus_tour_simulation.launch.py
```

```
[INFO] [launch]: All log files can be found below /home/tushar45/.ros/log/2024-09-30-23-16-26-176376-LAPTOP-OQEBKQF1-2439
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [bi_agent-1]: process started with pid [2440]
[INFO] [bi_agent-2]: process started with pid [2442]
[INFO] [bi_agent-3]: process started with pid [2444]
[INFO] [ci_agent-4]: process started with pid [2446]
[INFO] [visitor_agent-5]: process started with pid [2448]
[INFO] [visitor_agent-6]: process started with pid [2450]
[bi_agent-2] [INFO] [1727718390.591432753] [bi_agent_lhc]: BI Agent for LHC is ready
[bi_agent-3] [INFO] [1727718390.592132671] [bi_agent_b1_hostel]: BI Agent for B1 Hostel is ready
[bi_agent-1] [INFO] [1727718390.594711032] [bi_agent_library]: BI Agent for Library is ready
[ci_agent-4] [INFO] [1727718390.595240981] [ci_agent]: CI Agent is ready
[visitor_agent-5] [INFO] [1727718390.600001284] [visitor_1]: Visitor wants to go to LHC
[visitor_agent-6] [INFO] [1727718390.651822829] [visitor_2]: Visitor wants to go to LHC
[visitor_agent-5] [INFO] [1727718395.574319912] [visitor_1]: Arrived at LHC
[visitor_agent-6] [INFO] [1727718395.635730226] [visitor_2]: Escort error: CI Agent is busy
[visitor_agent-5] [INFO] [1727718400.570817022] [visitor_1]: In meeting at LHC
[visitor_agent-5] [INFO] [1727718400.572288006] [visitor_1]: In meeting at LHC
[visitor_agent-5] [INFO] [1727718400.573613592] [visitor_1]: In meeting at LHC
[visitor_agent-6] [INFO] [1727718400.633787551] [visitor_2]: Escort error: CI Agent is busy
[ci_agent-4] [INFO] [1727718405.573483969] [ci_agent]: CI Agent is now available
[visitor_agent-6] [INFO] [1727718405.636012273] [visitor_2]: Arrived at LHC
[visitor_agent-6] [INFO] [1727718410.634932071] [visitor_2]: In meeting at LHC
[visitor_agent-6] [INFO] [1727718410.635903656] [visitor_2]: In meeting at LHC
[visitor_agent-6] [INFO] [1727718410.636691198] [visitor_2]: In meeting at LHC
[ci_agent-4] [INFO] [1727718415.635951485] [ci_agent]: CI Agent is now available
```

You can see how the simulation is working.

Ci_agent

```
tushar45@LAPTOP-OQEBKQF1:~/ros2_ws/src/campus_virtual_tour$ ros2 run campus_virtual_tour ci_agent
[INFO] [1727719750.105914810] [ci_agent]: CI Agent is ready
[INFO] [1727719764.937030044] [ci_agent]: CI Agent is now available
```

```
[visitor_agent-6] [INFO] [1727719754.932355821] [visitor_2]: Escort error: CI Agent is busy
[visitor_agent-6] [INFO] [1727719754.940608546] [visitor_2]: Arrived at LHC
[ci_agent-4] [INFO] [1727719758.515026359] [ci_agent]: CI Agent is now available
[visitor_agent-6] [INFO] [1727719759.931613431] [visitor_2]: In meeting at LHC
[visitor_agent-6] [INFO] [1727719759.933274995] [visitor_2]: In meeting at LHC
[visitor_agent-6] [INFO] [1727719759.934942059] [visitor_2]: In meeting at LHC
```

We can see when the agent is free and when at a meeting. Escort error when busy.

Bi_agent

```
[visitor_agent-6] [INFO] [1727719860.277480618] [visitor_2]: Visitor wants to go to LHC
[visitor_agent-5] [INFO] [1727719860.283172772] [visitor_1]: Visitor wants to go to Library
[bi_agent-3] [INFO] [1727719860.284932908] [bi_agent_b1_hostel]: BI Agent for B1 Hostel is ready
[ci_agent-4] [INFO] [1727719860.286281915] [ci_agent]: CI Agent is ready
[bi_agent-1] [INFO] [1727719860.290532647] [bi_agent_library]: BI Agent for Library is ready
```

```
tushar45@LAPTOP-OQEBKQF1:~/ros2_ws/src/campus_virtual_tour$ ros2 run campus_virtual_tour bi_agent
[INFO] [1727719861.235003643] [bi_agent]: BI Agent for Library is ready
```

We can see how bi_agents are ready.

Visitor

```
tushar45@LAPTOP-OQEBKQF1:~/ros2_ws/src/campus_virtual_tour$ ros2 run campus_virtual_tour visitor_agent
[INFO] [1727720039.588938349] [visitor_agent]: Visitor wants to go to Library
[INFO] [1727720044.575189973] [visitor_agent]: Arrived at Library
[INFO] [1727720049.577676415] [visitor_agent]: In meeting at Library
[INFO] [1727720049.579118599] [visitor_agent]: In meeting at Library
[INFO] [1727720049.580509449] [visitor_agent]: In meeting at Library
```

```
[visitor_agent-5] [INFO] [1727719980.536020882] [visitor_1]: Visitor wants to go to B1 Hostel
[visitor_agent-6] [INFO] [1727719985.514821513] [visitor_2]: Arrived at Library
[visitor_agent-5] [INFO] [1727719985.515315920] [visitor_1]: Escort error: CI Agent is busy
[visitor_agent-6] [INFO] [1727719990.517337324] [visitor_2]: In meeting at Library
[visitor_agent-6] [INFO] [1727719990.518644291] [visitor_2]: In meeting at Library
[visitor_agent-6] [INFO] [1727719990.519702645] [visitor_2]: In meeting at Library
[visitor_agent-5] [INFO] [1727719990.521727489] [visitor_1]: Escort error: CI Agent is busy
```

The visitor visits a place and requests a bi_aagent.
When available, meetings are held.
Else, Escort error.

Rqt_graph