Introduction to Intelligent Autonomous Systems – LIACD 2025-26
Assignment 1 – list of proposed themes

A. Assignment: Multi-Agent Smart Traffic Management and Emergency Vehicle Routing System

Objective:

Design and implement a decentralized traffic management system using Multi-Agent Systems (MAS) with **SPADE**. The system will simulate an urban environment where multiple autonomous agents (traffic lights, vehicles, and emergency responders) collaborate to reduce congestion, optimize traffic flow, and ensure priority routing for emergency vehicles—all without relying on a central traffic control unit.

Problem Scenario:

In modern cities, centralized traffic control systems often struggle with scalability, failures, or communication breakdowns during peak hours or emergencies. A decentralized, agent-based approach can provide resilience and adaptability. Agents representing traffic lights, vehicles, and emergency responders must coordinate in real-time to manage congestion, reroute traffic, and guarantee that ambulances, fire trucks, or police vehicles can reach their destinations quickly and safely. Each agent must act autonomously while sharing information with peers to optimize the overall system.

Key Features of the Assignment:

1. **Traffic Light Agents:**
   Each traffic light is an independent agent that controls its intersection. It adapts signal timing based on local traffic density and communicates with neighboring intersections to balance flow.
2. **Vehicle Agents:**
   Vehicles are modeled as agents navigating the city grid. They choose routes based on traffic conditions, fuel/time constraints, and updates received from other agents.
3. **Emergency Vehicle Agents:**
   Special agents (ambulances, fire trucks, police) require priority routing. Other agents must collaborate to clear paths and dynamically adjust signals to ensure rapid passage.
4. **Resource Management:**
   Agents must manage limited resources such as fuel (for vehicles), waiting time (for drivers), and road capacity. Efficient use of these resources is critical to system performance.
5. **Decentralized Coordination:**
   No central controller exists. Agents exchange information peer-to-peer about congestion, accidents, or blocked roads. This prevents bottlenecks and increases robustness.

6. **Dynamic Environment:**
   The city environment changes over time (e.g., accidents, road closures, sudden traffic surges). Agents must adapt routes and signal timings dynamically.

7. **Collaboration Protocols:**
   Implement protocols such as the **Contract Net Protocol**, where traffic lights can delegate rerouting tasks or vehicles can negotiate alternative paths when congestion occurs.

8. **Performance Metrics:**
   The system should report on:
   ○ Average travel time of vehicles
   ○ Emergency vehicle response time
   ○ Fuel/time efficiency
   ○ Congestion reduction compared to baseline
   ○ Effectiveness of decentralized collaboration

9. **Agent Specialization:**
   To increase complexity, agents can have specialized roles:
   ○ **Intersection Managers:** Traffic light agents optimizing flow locally.
   ○ **Regular Vehicles:** Agents balancing personal efficiency with system-wide cooperation.
   ○ **Emergency Vehicles:** Agents with highest priority, requiring system-wide coordination.
   ○ **Incident Reporters:** Agents that detect and broadcast accidents or hazards.

Suggested Development Phases:

Week 1-2:

- **Introduction to MAS and SPADE:** Learn basics of agent creation, messaging, and autonomy.
- **System Design:** Define agent types (traffic lights, vehicles, emergency responders), communication protocols, and the city grid environment.
- **Initial Setup:** Implement basic agents that can move (vehicles) or control signals (traffic lights).

Week 3:

- **Agent Communication:** Implement decentralized communication so agents can share congestion data, blocked roads, or priority requests.
- **Basic Routing:** Vehicles navigate from origin to destination, reporting delays and adapting routes.

Week 4:

- **Resource Management:** Introduce fuel/time constraints for vehicles and capacity constraints for roads.
- **Dynamic Environment:** Add events such as accidents or road closures requiring agents to re-plan routes.

Week 5:

- **Advanced Collaboration:** Implement Contract Net Protocol for rerouting and coordination.
- **Emergency Vehicle Priority:** Ensure emergency vehicles can request priority passage, with traffic lights and vehicles adapting accordingly.

Week 6:

- **User Interface and Visualization:** Create a simple interface showing the city grid, traffic flow, congestion, and emergency vehicle routing.
- **Testing and Evaluation:** Run simulations under different traffic loads, accident scenarios, and emergency events. Evaluate performance using defined metrics.

B. Assignment: Multi-Agent Smart Energy Grid Management System

Objective:

Design and implement a decentralized smart energy grid management system using **SPADE**. The system will simulate a city-wide power grid where multiple agents (representing households, renewable energy producers, and grid nodes) collaborate to balance energy demand and supply in real-time, ensuring resilience and efficiency without relying on a central controller.

Problem Scenario:

Modern energy grids face challenges such as fluctuating renewable energy production (solar, wind), unpredictable consumption patterns, and risks of centralized system failures. A decentralized, agent-based approach allows local decision-making and peer-to-peer coordination. Agents representing households, renewable producers, and substations must negotiate energy exchanges, manage storage, and adapt to dynamic conditions (e.g., sudden demand spikes, equipment failures). The system should maximize efficiency, minimize blackouts, and ensure fair distribution of resources.

Key Features of the Assignment:

1. **Household Agents:**
   Represent consumers with varying energy demands. Some may also have solar panels or batteries, making them “prosumers” (producers + consumers).
2. **Producer Agents:**
   Represent renewable energy sources (solar farms, wind turbines). Their output fluctuates with time and conditions.
3. **Grid Node Agents:**
   Act as local hubs that balance supply and demand in their region. They coordinate with neighboring nodes to reroute energy when needed.

4. **Resource Management:**
   Agents must manage limited resources such as stored energy, production capacity, and transmission bandwidth.
5. **Decentralized Coordination:**
   No central grid operator exists. Agents negotiate energy trades peer-to-peer, ensuring local balance and system-wide stability.
6. **Dynamic Environment:**
   Energy demand and production vary over time (e.g., peak hours, cloudy days). Unexpected events (e.g., generator failure) force agents to adapt.
7. **Collaboration Protocols:**
   Implement protocols such as **Contract Net Protocol** for energy trading, where agents can auction surplus energy or request additional supply.
8. **Performance Metrics:**
   The system should report on:
   - Percentage of demand met
   - Energy wasted (unused surplus)
   - Fairness of distribution among households
   - Efficiency of decentralized coordination
9. **Agent Specialization:**
   - **Consumers/Prosumers:** Manage demand and storage.
   - **Producers:** Generate renewable energy with variability.
   - **Grid Nodes:** Balance local supply/demand and reroute energy.
   - **Storage Managers:** Handle batteries or other storage systems.

Suggested Development Phases:

Week 1-2:

- **Introduction to MAS and SPADE:** Learn basics of agent creation and communication.
- **System Design:** Define agent types, communication protocols, and the grid environment.
- **Initial Setup:** Implement basic agents that can consume or produce energy.

Week 3:

- **Agent Communication:** Implement decentralized negotiation for energy exchange.
- **Basic Balancing:** Agents share surplus/deficit information and trade energy.

Week 4:

- **Resource Management:** Add storage systems and constraints (battery capacity, transmission limits).
- **Dynamic Environment:** Introduce variable demand and renewable production.

Week 5:

- **Advanced Collaboration:** Implement Contract Net Protocol for energy trading.

- **Grid Node Coordination:** Nodes collaborate to reroute energy when local balance fails.

Week 6:

- **User Interface and Visualization:** Create a dashboard showing energy flows, demand vs. supply, and agent interactions.
- **Testing and Evaluation:** Simulate peak demand, renewable fluctuations, and failures. Evaluate performance with defined metrics.

C. Assignment: Multi-Agent Precision Agriculture and Farm Resource Management System

Objective:

Design and implement a decentralized precision agriculture system using **SPADE**. The system will simulate a smart farm where multiple agents (drones, irrigation units, soil sensors, and logistics units) collaborate to monitor crop health, optimize water and fertilizer usage, and manage harvesting tasks in real-time without a central controller.

Problem Scenario:

Modern farms face challenges such as unpredictable weather, limited water resources, and the need for sustainable practices. Centralized farm management systems can be vulnerable to failures or lack adaptability. A decentralized, agent-based approach allows autonomous decision-making and collaboration. Agents representing drones, irrigation systems, soil sensors, and harvesters must coordinate to monitor conditions, allocate resources efficiently, and adapt to dynamic environmental changes (e.g., drought, pest outbreaks).

Key Features of the Assignment:

1. **Drone Agents:**
   Fly over fields to monitor crop health, detect pests, and map areas needing attention.
2. **Soil Sensor Agents:**
   Provide localized data on soil moisture, nutrient levels, and temperature.
3. **Irrigation Agents:**
   Control water distribution in specific zones, adjusting based on sensor and drone data.
4. **Logistics/Harvester Agents:**
   Manage harvesting schedules, transport crops, and allocate machinery efficiently.
5. **Resource Management:**
   Agents must manage limited resources such as water, fertilizer, and fuel for machinery.
6. **Decentralized Coordination:**
   No central farm manager exists. Agents exchange information peer-to-peer to optimize irrigation, fertilization, and harvesting.

7. **Dynamic Environment:**
   Weather changes, pest outbreaks, or equipment failures force agents to adapt their strategies.

8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a drone detecting a dry patch can request irrigation agents to respond).

9. **Performance Metrics:**
   - Crop yield achieved
   - Water and fertilizer efficiency
   - Time taken for harvesting
   - Responsiveness to environmental changes
   - Effectiveness of collaboration

10. **Agent Specialization:**
    - **Monitoring Agents (drones, sensors):** Detect issues and share data.
    - **Irrigation Agents:** Manage water distribution.
    - **Fertilizer/Logistics Agents:** Deliver resources where needed.
    - **Harvesters:** Execute harvesting tasks.

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and farm environment.
- Initial setup: implement basic agents (drones, sensors, irrigation).

Week 3:

- Implement communication between agents (e.g., drones reporting dry zones to irrigation agents).
- Basic irrigation and monitoring logic.

Week 4:

- Add resource constraints (limited water/fertilizer).
- Introduce dynamic events (weather changes, pest outbreaks).
- Agents adapt strategies accordingly.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add logistics/harvester agents for crop collection and transport.

Week 6:

- Build a visualization interface showing farm layout, agent actions, and crop health.
- Test under different scenarios (drought, pest outbreak, resource shortages).

- Evaluate performance with defined metrics.

D. Assignment: Multi-Agent Hospital Resource Allocation and Emergency Response System

Objective:

Design and implement a decentralized hospital resource allocation system using **SPADE**. The system will simulate a healthcare network where multiple agents (hospitals, ambulances, doctors, and supply units) collaborate to allocate resources, manage patient flow, and respond to emergencies in real-time without a central command.

Problem Scenario:

Hospitals often face sudden surges in patient demand (e.g., during epidemics, accidents, or disasters). Centralized systems can become bottlenecks, leading to delays in treatment. A decentralized, agent-based system allows hospitals, ambulances, and medical staff to coordinate autonomously. Agents must share information about available beds, medical supplies, and staff capacity, while ensuring patients are routed efficiently and critical cases are prioritized.

Key Features of the Assignment:

1. **Hospital Agents:**
   Represent hospitals with limited beds, staff, and supplies. They must decide whether to admit patients or redirect them.
2. **Ambulance Agents:**
   Transport patients and decide which hospital to go to based on availability and proximity.
3. **Doctor/Nurse Agents:**
   Represent medical staff with limited capacity. They prioritize patients based on severity.
4. **Supply Agents:**
   Manage distribution of critical resources (oxygen, medicine, PPE) across hospitals.
5. **Resource Management:**
   Agents must manage limited beds, staff hours, and supplies, balancing efficiency and fairness.
6. **Decentralized Coordination:**
   No central health authority exists. Hospitals and ambulances exchange information peer-to-peer to optimize patient routing and resource allocation.
7. **Dynamic Environment:**
   Patient arrivals are unpredictable. Sudden surges (e.g., epidemic spikes) or supply shortages force agents to adapt.

8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a hospital unable to admit a patient negotiates with others to find a bed).

9. **Performance Metrics:**
   - Number of patients successfully treated
   - Average waiting/transport time
   - Resource utilization (beds, staff, supplies)
   - Fairness of distribution across hospitals
   - Responsiveness to emergencies

10. **Agent Specialization:**
    - **Hospitals:** Manage admissions and resources
    - **Ambulances:** Route patients efficiently
    - **Medical Staff:** Prioritize and treat patients
    - **Suppliers:** Deliver resources where needed

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and hospital network environment.
- Initial setup: implement basic hospital and ambulance agents.

Week 3:

- Implement communication between hospitals and ambulances.
- Basic patient routing and admission logic.

Week 4:

- Add resource constraints (beds, staff, supplies).
- Introduce dynamic patient arrivals and emergencies.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add supply agents and staff prioritization logic.

Week 6:

- Build a visualization interface showing hospitals, ambulances, patients, and resources.
- Test under different scenarios (epidemic surge, supply shortage, mass accident).
- Evaluate performance with defined metrics.

E. Assignment: Multi-Agent Decentralized Supply Chain and Delivery Optimization System

Objective:

Design and implement a decentralized supply chain and delivery optimization system using **SPADE**. The system will simulate a network of warehouses, delivery vehicles, and retail stores where multiple agents collaborate to manage inventory, fulfill orders, and optimize delivery routes in real-time without a central logistics controller.

Problem Scenario:

In global and urban supply chains, centralized logistics systems can become bottlenecks or fail under disruptions (e.g., strikes, traffic jams, or sudden demand spikes). A decentralized, agent-based approach allows warehouses, vehicles, and stores to negotiate and adapt autonomously. Agents must coordinate to ensure timely deliveries, minimize costs, and handle dynamic conditions such as traffic delays, stock shortages, or urgent orders.

Key Features of the Assignment:

1. **Warehouse Agents:**
   Manage stock levels, receive restocking requests, and negotiate with other warehouses or suppliers to balance inventory.
2. **Vehicle Agents:**
   Represent delivery trucks or vans. They plan routes, manage fuel/time constraints, and adapt to traffic or road closures.
3. **Retail Store Agents:**
   Place orders based on customer demand, negotiate with warehouses for fulfillment, and track delivery status.
4. **Supplier Agents:**
   Provide raw materials or goods to warehouses. They manage production capacity and delivery schedules.
5. **Resource Management:**
   Agents must manage limited resources such as fuel, delivery time windows, storage capacity, and product availability.
6. **Decentralized Coordination:**
   No central logistics hub exists. Agents negotiate peer-to-peer to fulfill orders, reroute deliveries, and share information about stock and traffic.
7. **Dynamic Environment:**
   Traffic jams, vehicle breakdowns, or sudden demand surges force agents to adapt their plans.

8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a warehouse unable to fulfill an order negotiates with another warehouse or vehicle).

9. **Performance Metrics:**
   - Order fulfillment rate
   - Average delivery time
   - Fuel/resource efficiency
   - Inventory balance across warehouses
   - Effectiveness of decentralized collaboration

10. **Agent Specialization:**
    - **Warehouses:** Manage stock and negotiate supply.
    - **Vehicles:** Optimize delivery routes.
    - **Stores:** Generate demand and track fulfillment.
    - **Suppliers:** Replenish warehouses.

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and supply chain environment.
- Initial setup: implement basic warehouse and store agents with simple order placement and fulfillment.

Week 3:

- Implement communication between warehouses, stores, and vehicles.
- Basic delivery logic: vehicles transport goods from warehouses to stores.

Week 4:

- Add resource constraints (fuel, time windows, storage limits).
- Introduce dynamic events (traffic delays, urgent orders, stock shortages).
- Agents adapt routes and inventory strategies.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add supplier agents for restocking.
- Vehicles negotiate route adjustments when overloaded or delayed.

Week 6:

- Build a visualization interface showing warehouses, vehicles, stores, and delivery routes.
- Test under different scenarios (demand spikes, traffic disruptions, supply shortages).
- Evaluate performance with defined metrics.

F. Assignment: Multi-Agent Decentralized Waste Collection and Recycling System

Objective:

Design and implement a decentralized waste collection and recycling management system using **SPADE**. The system will simulate a smart city where multiple agents (waste bins, collection trucks, recycling centers, and citizens) collaborate to optimize waste collection, reduce overflow, and maximize recycling efficiency without relying on a central controller.

Problem Scenario:

Urban waste management is often centralized, leading to inefficiencies such as overflowing bins, unnecessary truck routes, and poor recycling rates. A decentralized, agent-based approach allows bins, trucks, and recycling centers to coordinate autonomously. Agents must share information about bin fill levels, truck capacity, and recycling center availability, while adapting to dynamic conditions such as traffic congestion, sudden waste surges (festivals, events), or equipment breakdowns.

Key Features of the Assignment:

1. **Smart Bin Agents:**
   Each bin monitors its fill level and type of waste (general, recyclable, organic). When nearing capacity, it communicates with nearby trucks.
2. **Truck Agents:**
   Represent waste collection vehicles with limited capacity and fuel. They plan routes dynamically, responding to bin requests and recycling center availability.
3. **Recycling Center Agents:**
   Manage intake capacity for different waste types. They negotiate with trucks to accept deliveries and balance loads across centers.
4. **Citizen Agents (Optional):**
   Generate waste at varying rates, simulating real-world variability in bin fill levels.
5. **Resource Management:**
   Agents must manage limited truck capacity, fuel, recycling center throughput, and time.
6. **Decentralized Coordination:**
   No central waste authority exists. Bins, trucks, and centers negotiate peer-to-peer to optimize collection and recycling.
7. **Dynamic Environment:**
   Events like festivals, traffic jams, or truck breakdowns force agents to adapt their plans.

8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a bin requests collection, trucks bid based on proximity and capacity).

9. **Performance Metrics:**
   - Percentage of bins collected before overflow
   - Recycling rate achieved
   - Average truck fuel/time efficiency
   - Responsiveness to dynamic events
   - Effectiveness of decentralized collaboration

10. **Agent Specialization:**
    - **Bins:** Monitor and request collection
    - **Trucks:** Plan and execute routes
    - **Recycling Centers:** Manage intake and processing
    - **Citizens (optional):** Generate waste dynamically

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and city environment.
- Initial setup: implement basic bin and truck agents.

Week 3:

- Implement communication between bins and trucks.
- Basic collection logic: trucks respond to bin requests.

Week 4:

- Add resource constraints (truck capacity, fuel, recycling center limits).
- Introduce dynamic events (traffic, surges in waste).
- Agents adapt routes and collection strategies.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add recycling center agents to balance intake.
- Trucks negotiate with centers for optimal delivery.

Week 6:

- Build a visualization interface showing city layout, bins, trucks, and recycling centers.
- Test under different scenarios (waste surges, traffic jams, equipment failures).
- Evaluate performance with defined metrics.

G. Assignment: Multi-Agent Decentralized Intrusion Detection and Cyber Defense System

Objective:

Design and implement a decentralized intrusion detection and cyber defense system using **SPADE**. The system will simulate a computer network where multiple agents (workstations, servers, firewalls, and monitoring nodes) collaborate to detect, analyze, and mitigate cyberattacks in real-time without relying on a central security controller.

Problem Scenario:

Traditional intrusion detection systems (IDS) often rely on centralized monitoring, which can become a single point of failure or bottleneck during large-scale cyberattacks. In a decentralized approach, agents embedded across the network (e.g., on servers, routers, or endpoints) must autonomously monitor traffic, detect anomalies, and share alerts with peers. Agents must coordinate to confirm threats, isolate compromised nodes, and adapt to evolving attack strategies such as distributed denial-of-service (DDoS), malware propagation, or insider threats.

Key Features of the Assignment:

1. **Node Agents (Workstations/Servers):**
   Monitor local activity (e.g., login attempts, unusual traffic). They raise alerts when suspicious behavior is detected.
2. **Firewall/Router Agents:**
   Control network traffic flow. They can block suspicious IPs or reroute traffic based on alerts from other agents.
3. **Monitoring Agents:**
   Aggregate and analyze alerts from multiple nodes to confirm attacks and reduce false positives.
4. **Incident Response Agents:**
   Take corrective actions such as isolating compromised nodes, deploying patches, or redistributing workloads.
5. **Resource Management:**
   Agents must balance detection accuracy with limited computational resources and communication overhead.
6. **Decentralized Coordination:**
   No central security hub exists. Agents exchange alerts and negotiate responses peer-to-peer to ensure resilience.
7. **Dynamic Environment:**
   Attack patterns evolve over time (e.g., stealthy malware, sudden DDoS spikes). Agents must adapt detection thresholds and strategies.

8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a node detecting suspicious traffic requests confirmation from nearby monitoring agents).

9. **Performance Metrics:**
   - Detection rate (true positives)
   - False positive/false negative rates
   - Response time to mitigate attacks
   - Network resilience (uptime maintained during attacks)
   - Efficiency of decentralized collaboration

10. **Agent Specialization:**
    - **Detection Agents:** Monitor and raise alerts
    - **Firewall Agents:** Block or reroute traffic
    - **Monitoring Agents:** Confirm and correlate alerts
    - **Response Agents:** Execute mitigation strategies

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and network topology.
- Initial setup: implement basic detection agents that can identify simple anomalies.

Week 3:

- Implement communication between agents (alert sharing).
- Basic collaborative detection: multiple agents confirm suspicious activity.

Week 4:

- Add resource constraints (bandwidth, CPU usage).
- Introduce dynamic attack scenarios (e.g., brute force login attempts, malware spread).
- Agents adapt thresholds and detection strategies.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add firewall and response agents to actively mitigate threats.
- Agents negotiate responsibilities (e.g., which firewall blocks which IP).

Week 6:

- Build a visualization interface showing network topology, agent alerts, and attack/defense dynamics.
- Test under different attack scenarios (DDoS, malware, insider threat).
- Evaluate performance with defined metrics.

H. Assignment: Multi-Agent Planetary Exploration and Resource Mapping System

Objective:

Design and implement a decentralized planetary exploration system using **SPADE**. The system will simulate a team of autonomous rovers and drones deployed on a distant planet or moon. These agents must collaborate to explore terrain, map resources, and adapt to hazards in real-time without relying on a central mission control.

Problem Scenario:

In extraterrestrial environments (e.g., Mars, the Moon, or asteroids), communication with Earth is delayed and centralized control is impractical. Exploration and resource mapping must be carried out autonomously by a team of agents. Each rover or drone must independently navigate, detect resources (e.g., water ice, minerals), and share findings with peers. The agents must coordinate to maximize coverage, avoid redundant exploration, and adapt to dynamic hazards such as dust storms, terrain obstacles, or equipment malfunctions.

Key Features of the Assignment:

1. **Rover Agents:**
   Ground-based explorers that navigate terrain, analyze soil samples, and detect resources. Limited by battery power and mobility constraints.

2. **Drone Agents:**
   Aerial scouts that map larger areas quickly, identify points of interest, and relay data to rovers.

3. **Base Station Agents:**
   Represent local hubs that store collected data, recharge rovers/drones, and coordinate logistics. They are not central controllers but peers in the system.

4. **Resource Management:**
   Agents must manage limited energy, sensor usage, and communication bandwidth. Efficient planning is critical to mission success.

5. **Decentralized Coordination:**
   No central mission control exists. Agents exchange information peer-to-peer to divide exploration zones, share resource discoveries, and avoid duplication.

6. **Dynamic Environment:**
   Hazards such as dust storms, terrain blockages, or rover malfunctions force agents to adapt their plans.

7. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a drone detecting a promising site requests a rover to investigate).

8. **Performance Metrics:**
    - Percentage of terrain mapped
    - Number and accuracy of resource discoveries
    - Energy efficiency (battery usage)
    - Mission duration before agent failure
    - Effectiveness of decentralized collaboration
9. **Agent Specialization:**
    - **Rovers:** Ground exploration and sampling
    - **Drones:** Aerial mapping and scouting
    - **Base Stations:** Data aggregation and recharging logistics

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and planetary environment.
- Initial setup: implement basic rover and drone agents with simple movement and sensing.

Week 3:

- Implement communication between rovers and drones.
- Basic exploration logic: agents divide terrain and avoid overlap.

Week 4:

- Add resource constraints (battery, sensor usage).
- Introduce dynamic hazards (blocked paths, dust storms).
- Agents adapt exploration strategies.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add base station agents for data storage and recharging.
- Agents negotiate task delegation (e.g., drone scouts, rover investigates).

Week 6:

- Build a visualization interface showing planetary terrain, agent movements, and discovered resources.
- Test under different scenarios (hazards, resource distribution, agent failures).
- Evaluate performance with defined metrics.

I. Assignment: Multi-Agent Decentralized Urban Water Distribution and Leak Management System

Objective:

Design and implement a decentralized water distribution management system using **SPADE**. The system will simulate a city's water network where multiple agents (pumping stations, reservoirs, sensors, and repair crews) collaborate to ensure continuous water supply, detect leaks, and adapt to disruptions in real-time without relying on a central authority.

Problem Scenario:

Urban water networks are critical infrastructure, but centralized control systems can fail during disasters (earthquakes, floods, cyberattacks) or simply struggle with scale. A decentralized, agent-based approach allows local nodes to autonomously monitor, detect, and respond to issues. Agents representing pumps, reservoirs, sensors, and repair crews must coordinate to balance supply and demand, detect leaks, and reroute water when parts of the system are compromised.

Key Features of the Assignment:

1. **Pump Station Agents:**
   Control water flow into the network. They adjust output based on demand and feedback from sensors.

2. **Reservoir Agents:**
   Manage water storage levels, negotiate with pumps and other reservoirs to balance supply.

3. **Sensor Agents:**
   Detect leaks, pressure drops, or contamination. They raise alerts and share data with nearby agents.

4. **Repair Crew Agents:**
   Respond to leak alerts, prioritize repairs, and manage limited resources (time, tools, personnel).

5. **Resource Management:**
   Agents must manage limited water supply, pumping capacity, and repair crew availability.

6. **Decentralized Coordination:**
   No central water authority exists. Agents negotiate peer-to-peer to reroute water, allocate crews, and balance demand.

7. **Dynamic Environment:**
   Pipe bursts, contamination events, or sudden demand spikes (e.g., during a fire) force agents to adapt.

8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a sensor detecting a leak requests repair crews, which bid based on proximity and availability).

9. **Performance Metrics:**
    - Percentage of demand met
    - Time to detect and repair leaks
    - Water loss due to leaks
    - Resource utilization (pumping energy, crew time)
    - Effectiveness of decentralized collaboration
10. **Agent Specialization:**
    - **Pumps:** Control flow
    - **Reservoirs:** Manage storage
    - **Sensors:** Detect anomalies
    - **Repair Crews:** Fix leaks and restore service

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and water network environment.
- Initial setup: implement basic pump, reservoir, and sensor agents.

Week 3:

- Implement communication between pumps, reservoirs, and sensors.
- Basic water flow balancing logic.

Week 4:

- Add resource constraints (limited pumping capacity, finite water supply).
- Introduce dynamic events (pipe bursts, demand spikes).
- Agents adapt flow and reroute water.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add repair crew agents to respond to leaks.
- Agents negotiate task assignments.

Week 6:

- Build a visualization interface showing the water network, flow, leaks, and repairs.
- Test under different scenarios (multiple leaks, contamination, high demand).
- Evaluate performance with defined metrics.

J. Assignment: Multi-Agent Decentralized Ride-Sharing and Fleet Optimization System

Objective:

Design and implement a decentralized ride-sharing system using **SPADE**. The system will simulate a city where multiple agents (passengers, vehicles, and dispatch hubs) collaborate to match rides, optimize routes, and balance fleet usage in real-time without relying on a central dispatcher.

Problem Scenario:

Traditional ride-sharing platforms rely on centralized servers to match passengers with drivers. This creates bottlenecks, single points of failure, and potential inefficiencies. In a decentralized approach, vehicles, passengers, and local hubs act as autonomous agents that negotiate directly. They must coordinate to minimize passenger waiting times, reduce empty vehicle mileage, and adapt to dynamic conditions such as traffic jams, sudden demand surges, or vehicle breakdowns.

Key Features of the Assignment:

1. **Passenger Agents:**
   Request rides, specify pickup and drop-off locations, and evaluate offers from nearby vehicles.
2. **Vehicle Agents:**
   Represent ride-sharing cars with limited seating and fuel. They negotiate with passengers, plan routes, and adapt to traffic conditions.
3. **Local Hub Agents (Optional):**
   Act as neighborhood-level coordinators that aggregate demand and help vehicles balance loads, but without central authority.
4. **Resource Management:**
   Agents must manage limited fuel, time, and seating capacity.
5. **Decentralized Coordination:**
   No central dispatcher exists. Vehicles and passengers negotiate directly, possibly through peer-to-peer bidding.
6. **Dynamic Environment:**
   Traffic congestion, sudden demand spikes (e.g., after a concert), or vehicle breakdowns force agents to adapt.
7. **Collaboration Protocols:**
   Use **Contract Net Protocol** for ride allocation (e.g., a passenger issues a request, vehicles bid based on proximity and availability).
8. **Performance Metrics:**
   - Average passenger waiting time
   - Fleet utilization (percentage of vehicles active)

  - Fuel/time efficiency
  - Ride-matching success rate
  - Effectiveness of decentralized collaboration

9. **Agent Specialization:**
   - **Passengers:** Generate ride requests
   - **Vehicles:** Fulfill requests and optimize routes
   - **Local Hubs (optional):** Facilitate neighborhood-level coordination

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and city environment.
- Initial setup: implement basic passenger and vehicle agents.

Week 3:

- Implement communication between passengers and vehicles.
- Basic ride-matching logic: vehicles respond to passenger requests.

Week 4:

- Add resource constraints (fuel, seating).
- Introduce dynamic events (traffic congestion, demand surges).
- Agents adapt routes and ride offers.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add optional hub agents for neighborhood-level optimization.
- Vehicles negotiate to swap or delegate rides.

Week 6:

- Build a visualization interface showing city map, vehicles, passengers, and routes.
- Test under different scenarios (rush hour, vehicle breakdowns, high demand).
- Evaluate performance with defined metrics.

K. Assignment: Multi-Agent Decentralized Adaptive Learning and Tutoring System

Objective:

Design and implement a decentralized adaptive learning system using **SPADE**. The system will simulate a network of students, tutors, and resource agents that collaborate to personalize learning paths, share knowledge, and optimize resource allocation in real-time without relying on a central learning management system.

Problem Scenario:

In large-scale education environments (e.g., online platforms, blended classrooms, or MOOCs), centralized systems often struggle to adapt to individual learner needs and can become bottlenecks. A decentralized, agent-based approach allows students, tutors, and resources to interact autonomously. Agents must coordinate to match learners with appropriate materials, provide peer-to-peer support, and adapt to dynamic conditions such as varying student progress, limited tutor availability, or sudden surges in demand for certain topics.

Key Features of the Assignment:

1. **Student Agents:**
   Represent learners with unique profiles (knowledge level, learning style, progress). They request resources, ask for help, and share knowledge with peers.
2. **Tutor Agents:**
   Represent human or AI tutors with limited availability. They provide explanations, feedback, and guidance, prioritizing students based on need.
3. **Resource Agents:**
   Manage educational materials (videos, exercises, quizzes). They recommend resources based on student profiles and progress.
4. **Peer Collaboration Agents (Optional):**
   Facilitate group learning by connecting students with complementary strengths.
5. **Resource Management:**
   Agents must manage limited tutor time, bandwidth for resources, and student attention spans.
6. **Decentralized Coordination:**
   No central LMS exists. Students, tutors, and resources negotiate directly to optimize learning outcomes.
7. **Dynamic Environment:**
   Students progress at different rates, tutors may become unavailable, and new topics may be introduced mid-course.
8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a student requests help, tutors bid based on availability and expertise).
9. **Performance Metrics:**
   - Student learning gains (knowledge improvement)
   - Average time to resolve learning difficulties
   - Tutor workload balance
   - Resource utilization efficiency
   - Effectiveness of decentralized collaboration

10. **Agent Specialization:**
   - ○ **Students:** Request and consume resources, share knowledge
   - ○ **Tutors:** Provide guidance and prioritize learners
   - ○ **Resources:** Recommend and deliver content
   - ○ **Peer Agents:** Enable collaborative learning

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and learning environment.
- Initial setup: implement basic student and resource agents.

Week 3:

- Implement communication between students and resources.
- Basic adaptive learning logic: resources recommend content based on student profiles.

Week 4:

- Add tutor agents with limited availability.
- Introduce dynamic events (students struggling, tutors unavailable).
- Agents adapt to ensure continuity of learning.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add peer collaboration agents for group learning.
- Agents negotiate task assignments (e.g., tutor vs. peer support).

Week 6:

- Build a visualization interface showing students, tutors, resources, and interactions.
- Test under different scenarios (high demand, uneven progress, tutor shortages).
- Evaluate performance with defined metrics.

L. Assignment: Multi-Agent Decentralized Wildlife Monitoring and Anti-Poaching System

Objective:

Design and implement a decentralized wildlife monitoring and anti-poaching system using **SPADE**. The system will simulate a protected reserve where multiple agents (drones, ranger patrols, sensor nodes, and animal trackers) collaborate to monitor wildlife populations, detect threats, and coordinate responses in real-time without relying on a central command center.

Problem Scenario:

Wildlife reserves face challenges such as poaching, habitat degradation, and the need to track animal movements across large, remote areas. Centralized monitoring systems are vulnerable to communication breakdowns and delays. A decentralized, agent-based approach allows drones, rangers, and sensors to act autonomously while sharing information peer-to-peer. Agents must coordinate to detect suspicious activity, track animal movements, and deploy rangers efficiently, all while managing limited resources like fuel, battery life, and patrol time.

Key Features of the Assignment:

1. **Drone Agents:**
   Patrol the skies, scan for animal herds or suspicious human activity, and relay information to ground agents. Limited by battery life and flight range.
2. **Ranger Agents:**
   Represent patrol units on the ground. They respond to alerts, investigate suspicious activity, and protect wildlife. Limited by fuel, time, and terrain accessibility.
3. **Sensor Node Agents:**
   Fixed sensors deployed across the reserve. They detect motion, sounds, or environmental changes (e.g., gunshots, vehicle noise) and raise alerts.
4. **Animal Tracker Agents:**
   Represent tagged animals with GPS collars. They provide movement data, helping agents predict migration patterns or detect unusual behavior (e.g., sudden stops indicating distress).
5. **Resource Management:**
   Agents must manage limited fuel, battery, and patrol time. Efficient allocation is critical to maximize coverage.
6. **Decentralized Coordination:**
   No central control tower exists. Agents negotiate directly to assign patrols, share alerts, and avoid redundant coverage.
7. **Dynamic Environment:**
   Poachers may move unpredictably, animals migrate, and weather conditions (storms, fog) affect visibility and mobility.
8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a sensor detects suspicious movement and requests drones or rangers to investigate).
9. **Performance Metrics:**
   - Percentage of threats detected and intercepted
   - Coverage of reserve area
   - Animal safety (number of animals tracked and protected)

  - Resource efficiency (fuel, battery usage)
  - Effectiveness of decentralized collaboration

10. **Agent Specialization:**
    - **Drones:** Aerial surveillance
    - **Rangers:** Ground response
    - **Sensors:** Passive detection
    - **Animal Trackers:** Wildlife monitoring

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and reserve environment.
- Initial setup: implement basic drone and sensor agents.

Week 3:

- Implement communication between drones, sensors, and rangers.
- Basic surveillance logic: drones patrol, sensors detect, rangers respond.

Week 4:

- Add resource constraints (battery, fuel, patrol time).
- Introduce dynamic events (poacher movement, animal migration).
- Agents adapt patrols and responses.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add animal tracker agents to simulate wildlife movement.
- Agents negotiate task assignments (e.g., ranger vs. drone response).

Week 6:

- Build a visualization interface showing reserve map, agents, animals, and alerts.
- Test under different scenarios (poaching attempts, migration, weather disruptions).
- Evaluate performance with defined metrics.

M. Assignment: Multi-Agent Decentralized Smart Factory Production and Maintenance System

Objective:

Design and implement a decentralized smart factory management system using **SPADE**. The system will simulate a factory floor where multiple agents (machines, robots, maintenance crews, and supply units) collaborate to schedule production tasks, manage resources, and handle equipment failures in real-time without relying on a central controller.

Problem Scenario:

Modern factories face challenges such as machine breakdowns, supply delays, and fluctuating production demands. Centralized scheduling systems can become bottlenecks or fail under disruptions. A decentralized, agent-based approach allows machines, robots, and maintenance crews to negotiate tasks autonomously. Agents must coordinate to ensure production targets are met, downtime is minimized, and resources are used efficiently, even in the face of unexpected failures or demand changes.

Key Features of the Assignment:

1. **Machine Agents:**
   Represent production machines with specific capabilities (e.g., cutting, assembling, packaging). They accept jobs, monitor their health, and request maintenance when needed.
2. **Robot/Worker Agents:**
   Handle material transport between machines, manage assembly tasks, and adapt to workload changes.
3. **Maintenance Crew Agents:**
   Respond to machine breakdowns or preventive maintenance requests. Limited by availability and repair time.
4. **Supply Agents:**
   Manage raw materials and deliver them to machines. They negotiate with machines to ensure timely supply.
5. **Resource Management:**
   Agents must manage limited machine capacity, raw materials, energy, and maintenance resources.
6. **Decentralized Coordination:**
   No central production planner exists. Agents negotiate peer-to-peer to allocate jobs, reroute tasks, and balance workloads.
7. **Dynamic Environment:**
   Machine breakdowns, supply delays, or sudden changes in production orders force agents to adapt.
8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a machine unable to complete a job requests another machine to take over).
9. **Performance Metrics:**
   ○ Production throughput (jobs completed)
   ○ Downtime due to failures
   ○ Resource utilization (machines, materials, energy)

- ○ Responsiveness to disruptions
- ○ Effectiveness of decentralized collaboration

10. **Agent Specialization:**
    - ○ **Machines:** Execute production tasks
    - ○ **Robots/Workers:** Transport and assist
    - ○ **Maintenance Crews:** Repair and maintain
    - ○ **Suppliers:** Deliver raw materials

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and factory environment.
- Initial setup: implement basic machine and supply agents.

Week 3:

- Implement communication between machines and suppliers.
- Basic job allocation and material delivery logic.

Week 4:

- Add resource constraints (machine capacity, material availability).
- Introduce dynamic events (machine breakdowns, supply delays).
- Agents adapt production schedules.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add maintenance crew agents to handle breakdowns.
- Agents negotiate task reallocation during disruptions.

Week 6:

- Build a visualization interface showing factory layout, machines, jobs, and resources.
- Test under different scenarios (high demand, multiple breakdowns, supply shortages).
- Evaluate performance with defined metrics.

N. Assignment: Multi-Agent Decentralized Urban Power Outage Response and Microgrid Coordination System

Objective:

Design and implement a decentralized outage response and microgrid coordination system using **SPADE**. The system will simulate a smart city where multiple agents (households, microgrids, repair crews, and renewable energy units) collaborate to restore power, share energy, and adapt to failures in real-time without relying on a central utility operator.

Problem Scenario:

Cities are increasingly vulnerable to large-scale power outages caused by storms, cyberattacks, or equipment failures. Centralized grid management can be slow to respond and fragile under stress. A decentralized, agent-based approach allows local microgrids, households, and repair crews to autonomously coordinate. Agents must detect outages, reroute power, share surplus energy, and prioritize critical facilities (e.g., hospitals, emergency shelters) while managing limited resources such as fuel, batteries, and repair capacity.

Key Features of the Assignment:

1. **Household Agents:**
   Represent homes or buildings with varying energy needs. Some may have solar panels or batteries, making them "prosumers."
2. **Microgrid Agents:**
   Manage clusters of households. They balance local supply and demand, negotiate with other microgrids, and reroute power when needed.
3. **Renewable Producer Agents:**
   Represent solar farms, wind turbines, or backup generators. Their output fluctuates with conditions.
4. **Repair Crew Agents:**
   Respond to outage reports, prioritize repairs, and manage limited tools, time, and mobility.
5. **Resource Management:**
   Agents must manage limited energy, storage, and repair resources. Efficient allocation is critical to minimize downtime.
6. **Decentralized Coordination:**
   No central utility operator exists. Agents negotiate peer-to-peer to share energy, reroute supply, and allocate repair crews.
7. **Dynamic Environment:**
   Outages spread unpredictably, renewable output fluctuates, and new failures may occur during repair.
8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a microgrid requests surplus energy, and nearby producers bid to supply it).
9. **Performance Metrics:**
   - Percentage of households with restored power
   - Time to restore critical facilities
   - Energy efficiency (minimizing waste)

    - Fairness of distribution across neighborhoods
    - Effectiveness of decentralized collaboration

10. **Agent Specialization:**
    - **Households/Prosumers:** Consume and share energy
    - **Microgrids:** Balance and reroute supply
    - **Producers:** Generate renewable energy
    - **Repair Crews:** Restore damaged infrastructure

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and city grid environment.
- Initial setup: implement basic household and microgrid agents.

Week 3:

- Implement communication between households and microgrids.
- Basic outage detection and reporting logic.

Week 4:

- Add resource constraints (battery capacity, renewable variability).
- Introduce dynamic events (storm damage, cascading failures).
- Agents adapt to reroute power.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add repair crew agents to restore damaged nodes.
- Agents negotiate task assignments and energy sharing.

Week 6:

- Build a visualization interface showing city map, microgrids, outages, and repairs.
- Test under different scenarios (widespread blackout, renewable fluctuations, multiple failures).
- Evaluate performance with defined metrics.

O. Assignment: Multi-Agent Decentralized Public Transportation Scheduling and Disruption Management System

Objective:

Design and implement a decentralized public transportation management system using **SPADE**. The system will simulate a city's bus and tram network where multiple agents (vehicles, stations, passengers, and maintenance crews) collaborate to optimize schedules, handle disruptions, and ensure efficient service delivery in real-time without relying on a central control center.

Problem Scenario:

Urban public transport systems often rely on centralized scheduling, which can fail or become inefficient during disruptions such as traffic jams, vehicle breakdowns, or sudden surges in passenger demand. A decentralized, agent-based approach allows buses, trams, stations, and passengers to negotiate autonomously. Agents must coordinate to minimize passenger waiting times, reroute vehicles during disruptions, and balance fleet usage across the network.

Key Features of the Assignment:

1. **Vehicle Agents (Buses/Trams):**
   Operate along routes, manage capacity, and adapt schedules based on real-time conditions. They negotiate with stations and other vehicles to avoid overcrowding.
2. **Station Agents:**
   Represent bus stops or tram stations. They monitor passenger queues, request additional vehicles when overcrowded, and share demand forecasts with nearby stations.
3. **Passenger Agents:**
   Generate travel requests with specific origins and destinations. They choose routes dynamically based on available vehicles and waiting times.
4. **Maintenance Crew Agents:**
   Respond to vehicle breakdowns, prioritize repairs, and manage limited resources (tools, time, personnel).
5. **Resource Management:**
   Agents must manage limited vehicle capacity, fuel/energy, and maintenance resources.
6. **Decentralized Coordination:**
   No central dispatcher exists. Vehicles, stations, and passengers negotiate directly to optimize service.
7. **Dynamic Environment:**
   Traffic congestion, breakdowns, or sudden demand spikes (e.g., after a concert) force agents to adapt.

8. **Collaboration Protocols:**
   Use **Contract Net Protocol** for task delegation (e.g., a station requests extra capacity, and nearby vehicles bid to reroute).

9. **Performance Metrics:**
   - Average passenger waiting time
   - Fleet utilization (percentage of vehicles active)
   - On-time performance of routes
   - Passenger satisfaction (successful trips completed)
   - Effectiveness of decentralized collaboration

10. **Agent Specialization:**
    - **Vehicles:** Transport passengers and adapt routes
    - **Stations:** Monitor demand and request service
    - **Passengers:** Generate travel requests and adapt choices
    - **Maintenance Crews:** Repair and restore vehicles

Suggested Development Phases:

Week 1-2:

- Introduction to MAS and SPADE.
- System design: define agent types, communication protocols, and city transport environment.
- Initial setup: implement basic vehicle and passenger agents.

Week 3:

- Implement communication between stations, vehicles, and passengers.
- Basic ride allocation logic: vehicles respond to station requests.

Week 4:

- Add resource constraints (vehicle capacity, fuel/energy).
- Introduce dynamic events (traffic congestion, breakdowns).
- Agents adapt schedules and routes.

Week 5:

- Implement advanced collaboration protocols (Contract Net Protocol).
- Add maintenance crew agents for breakdown response.
- Vehicles negotiate rerouting to balance demand.

Week 6:

- Build a visualization interface showing city map, vehicles, stations, and passenger flows.
- Test under different scenarios (rush hour, breakdowns, demand surges).
- Evaluate performance with defined metrics.