Autonomous Systems A LAB

Truck Platoon

Team 1

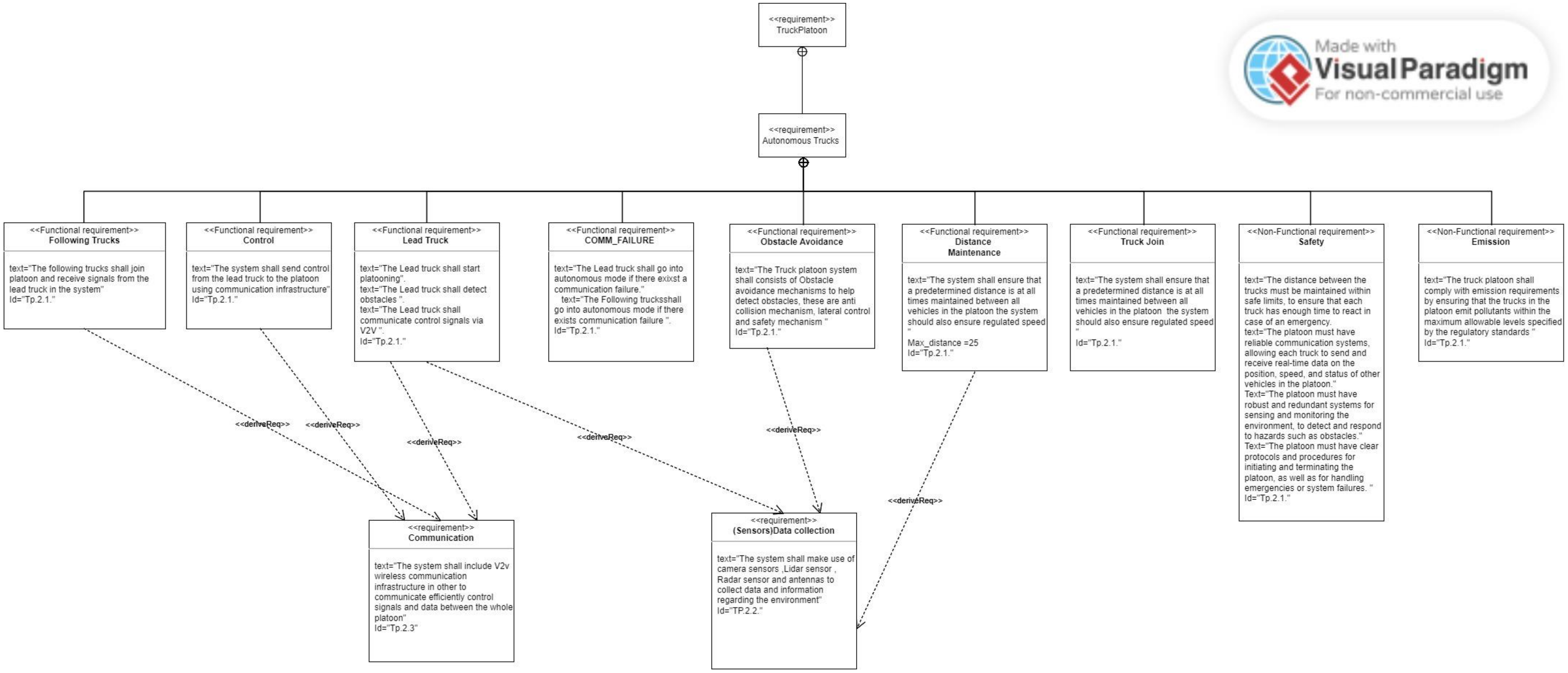
Presented By:

Kuye Doluwamu

Ashraf Siddiqi

Arsal Abbasi

System Requirement



<<requirement>>
TruckPlatoon

<<requirement>>
Autonomous Trucks

<<Functional requirement>>
Following Trucks

text="The following trucks shall join platoon and receive signals from the lead truck in the system"
Id="Tp.2.1."

<<Functional requirement>>
Control

text="The system shall send control from the lead truck to the platoon using communication infrastructure"
Id="Tp.2.1."

<<Functional requirement>>
Lead Truck

text="The Lead truck shall start platooning".
text="The Lead truck shall detect obstacles ".
text="The Lead truck shall communicate control signals via V2V ".
Id="Tp.2.1."

<<Functional requirement>>
COMM_FAILURE

text="The Lead truck shall go into autonomous mode if there exixst a communication failure."
text="The Following trucksshall go into autonomous mode if there exists communication failure ".
Id="Tp.2.1."

<<Functional requirement>>
Obstacle Avoidance

text="The Truck platoon system shall consists of Obstacle avoidance mechanisms to help detect obstacles, these are anti collision mechanism, lateral control and safety mechanism "
Id="Tp.2.1."

<<Functional requirement>>
Distance Maintenance

text="The system shall ensure that a predetermined distance is at all times maintained between all vehicles in the platoon the system should also ensure regulated speed "
Max_distance =25
Id="Tp.2.1."

<<Functional requirement>>
Truck Join

text="The system shall ensure that a predetermined distance is at all times maintained between all vehicles in the platoon the system should also ensure regulated speed "
Id="Tp.2.1."

<<Non-Functional requirement>>
Safety

text="The distance between the trucks must be maintained within safe limits, to ensure that each truck has enough time to react in case of an emergency.
text="The platoon must have reliable communication systems, allowing each truck to send and receive real-time data on the position, speed, and status of other vehicles in the platoon."
Text="The platoon must have robust and redundant systems for sensing and monitoring the environment, to detect and respond to hazards such as obstacles."
Text="The platoon must have clear protocols and procedures for initiating and terminating the platoon, as well as for handling emergencies or system failures. "
Id="Tp.2.1."

<<Non-Functional requirement>>
Emission

text="The truck platoon shall comply with emission requirements by ensuring that the trucks in the platoon emit pollutants within the maximum allowable levels specified by the regulatory standards "
Id="Tp.2.1."

<<requirement>>
Communication

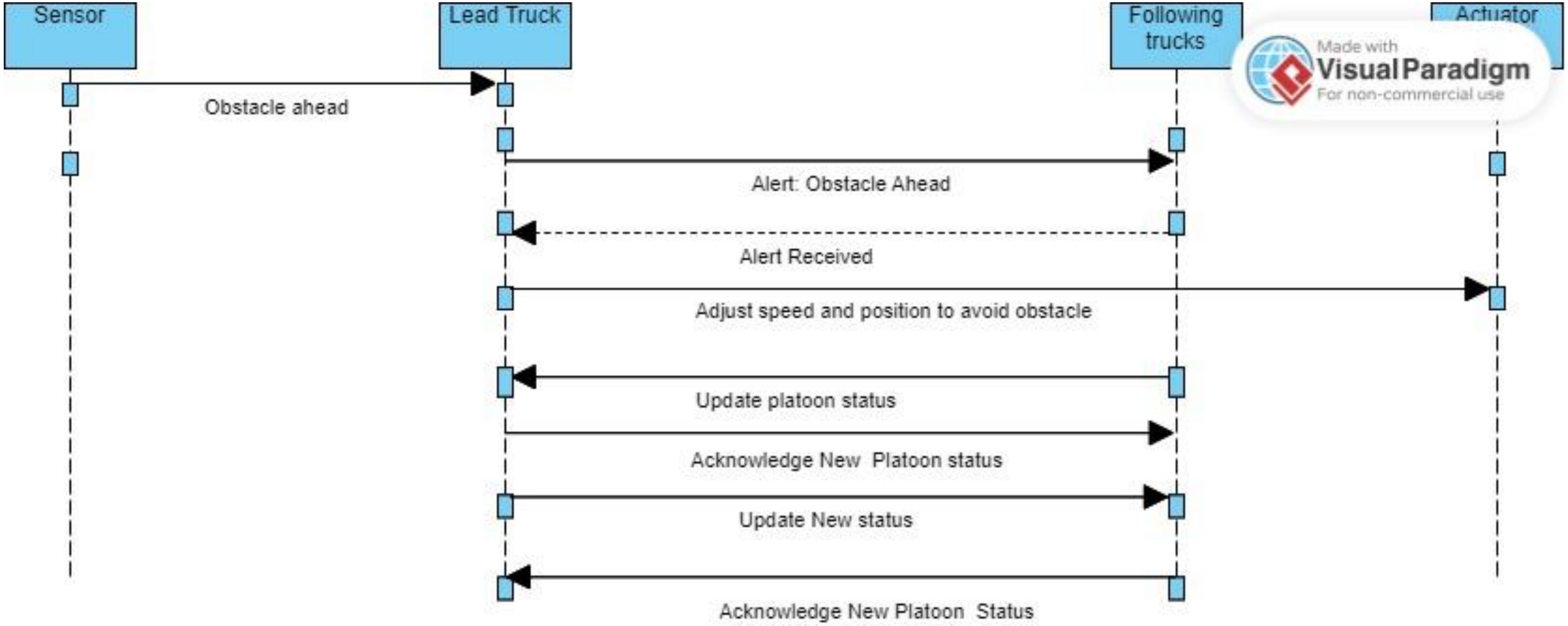
text="The system shall include V2v wireless communication infrastructure in other to communicate efficiently control signals and data between the whole platoon"
Id="Tp.2.3"

<<requirement>>
(Sensors)Data collection

text="The system shall make use of camera sensors ,Lidar sensor , Radar sensor and antennas to collect data and information regarding the environment"
Id="TP.2.2."

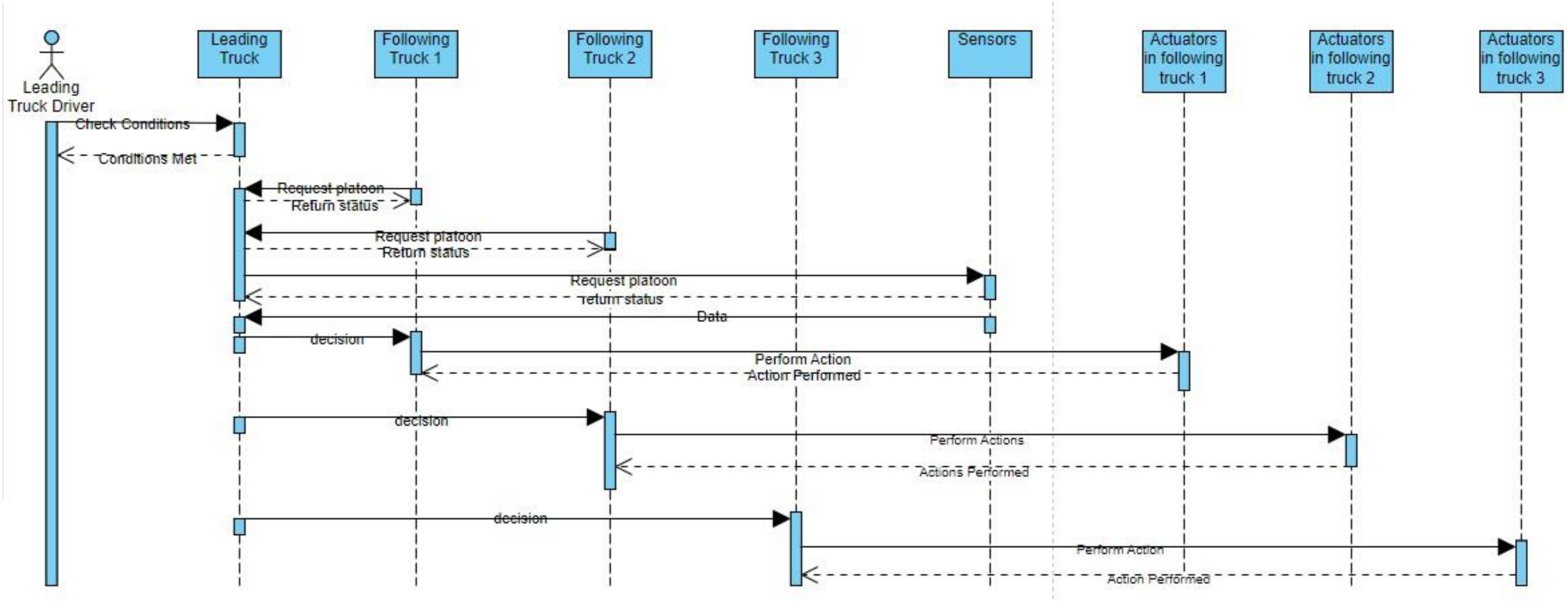
Obstacle Avoidance

- Sensor notifies lead truck about Obstacles
- Lead truck notifies Platoon
- Status Updated and platoon allignment


Sensor
Lead Truck
Following trucks
Actuator
Made with Visual Paradigm For non-commercial use
Obstacle ahead
Alert: Obstacle Ahead
Alert Received
Adjust speed and position to avoid obstacle
Update platoon status
Acknowledge New Platoon status
Update New status
Acknowledge New Platoon Status

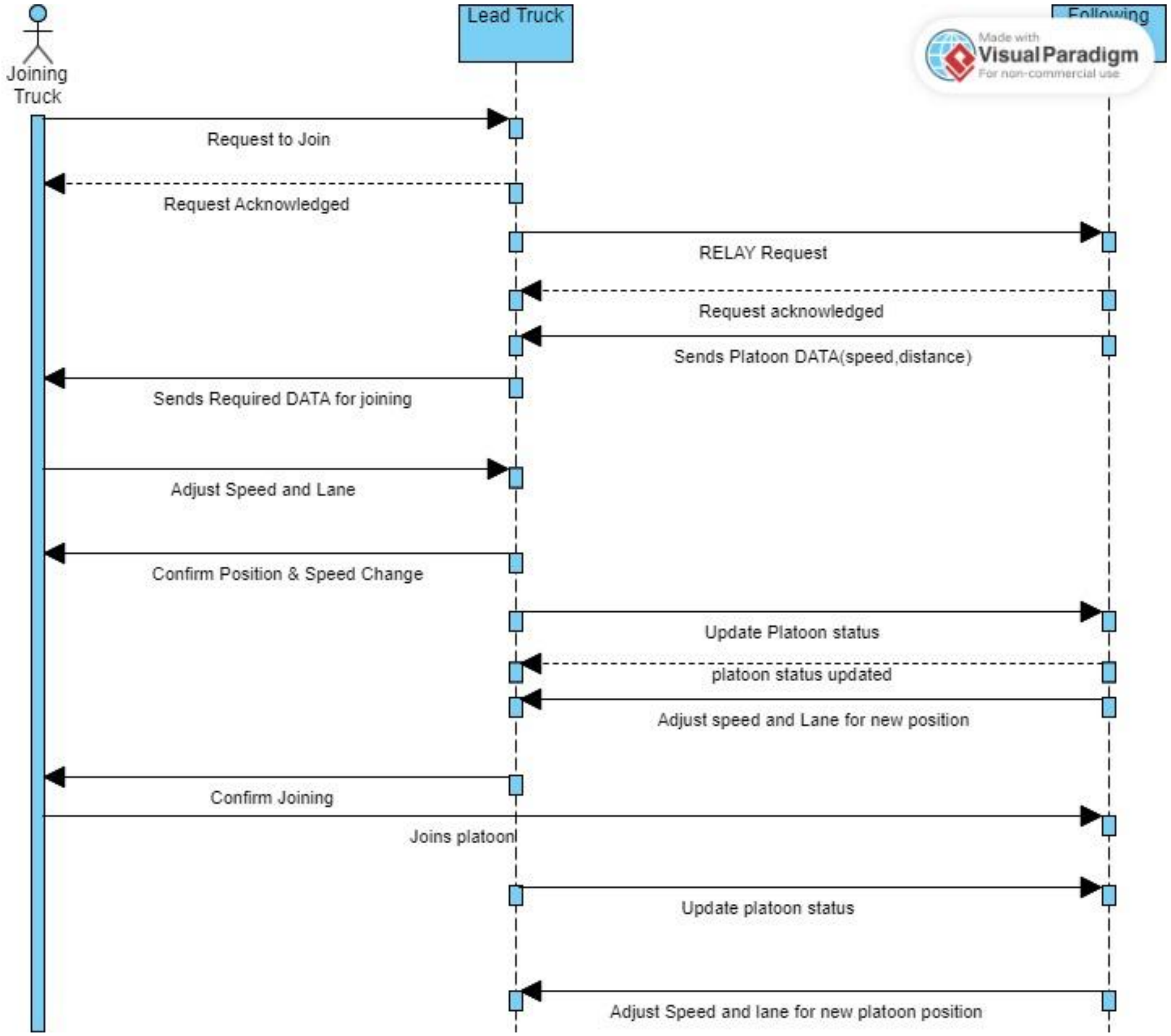
Maintaining Distance

- Lead truck receives platoon status.
- Decision is made based on current data
- Following truck performs actions based on the decision



Joining Truck

- Joining truck sends request
- Lead truck Notifies platoon.
- Joining data is sent
- Platoon status is Updated



Joining Truck

Lead Truck

Following

Request to Join

Request Acknowledged

RELAY Request

Request acknowledged

Sends Platoon DATA(speed,distance)

Sends Required DATA for joining

Adjust Speed and Lane

Confirm Position & Speed Change

Update Platoon status

platoon status updated

Adjust speed and Lane for new position

Confirm Joining

Joins platoon

Update platoon status

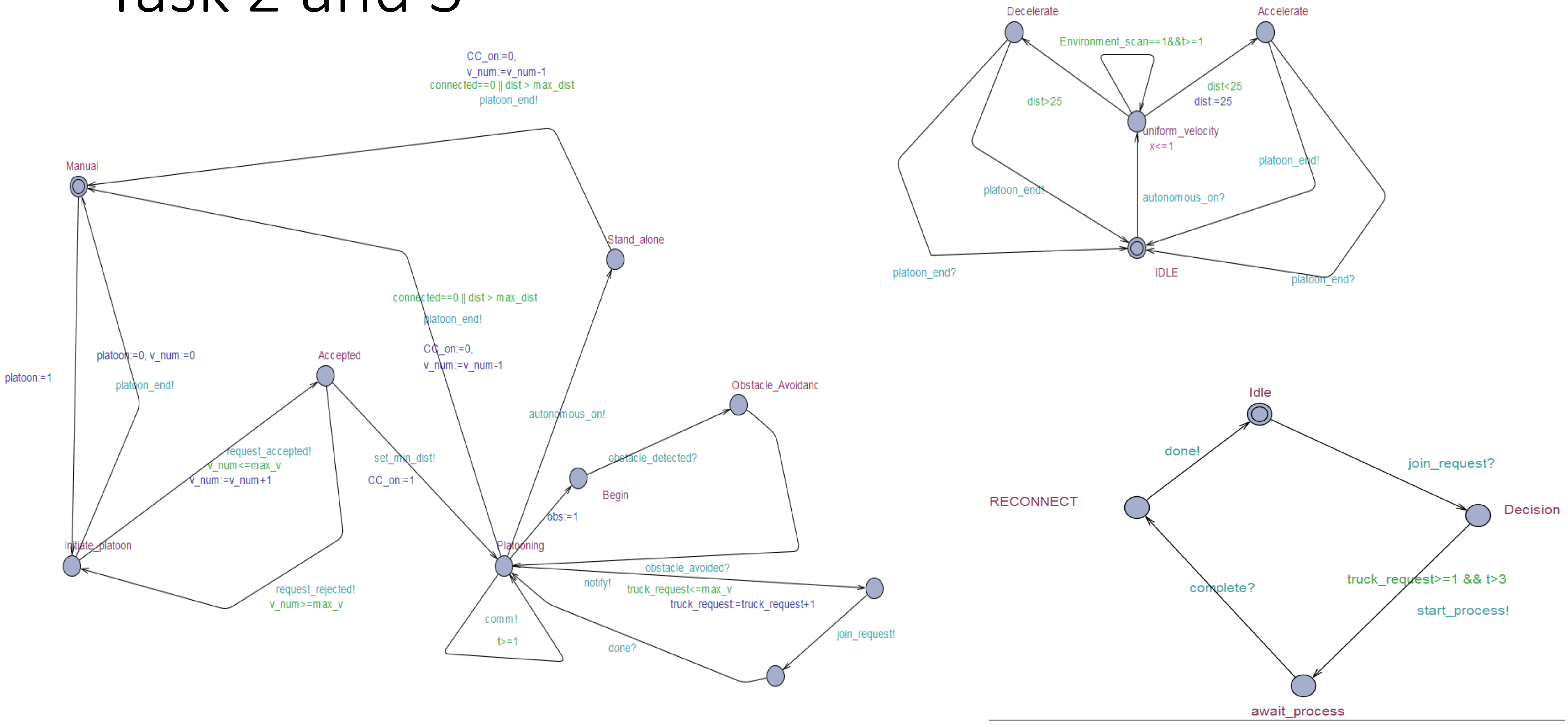
Adjust Speed and lane for new platoon position

Track 2 and 3

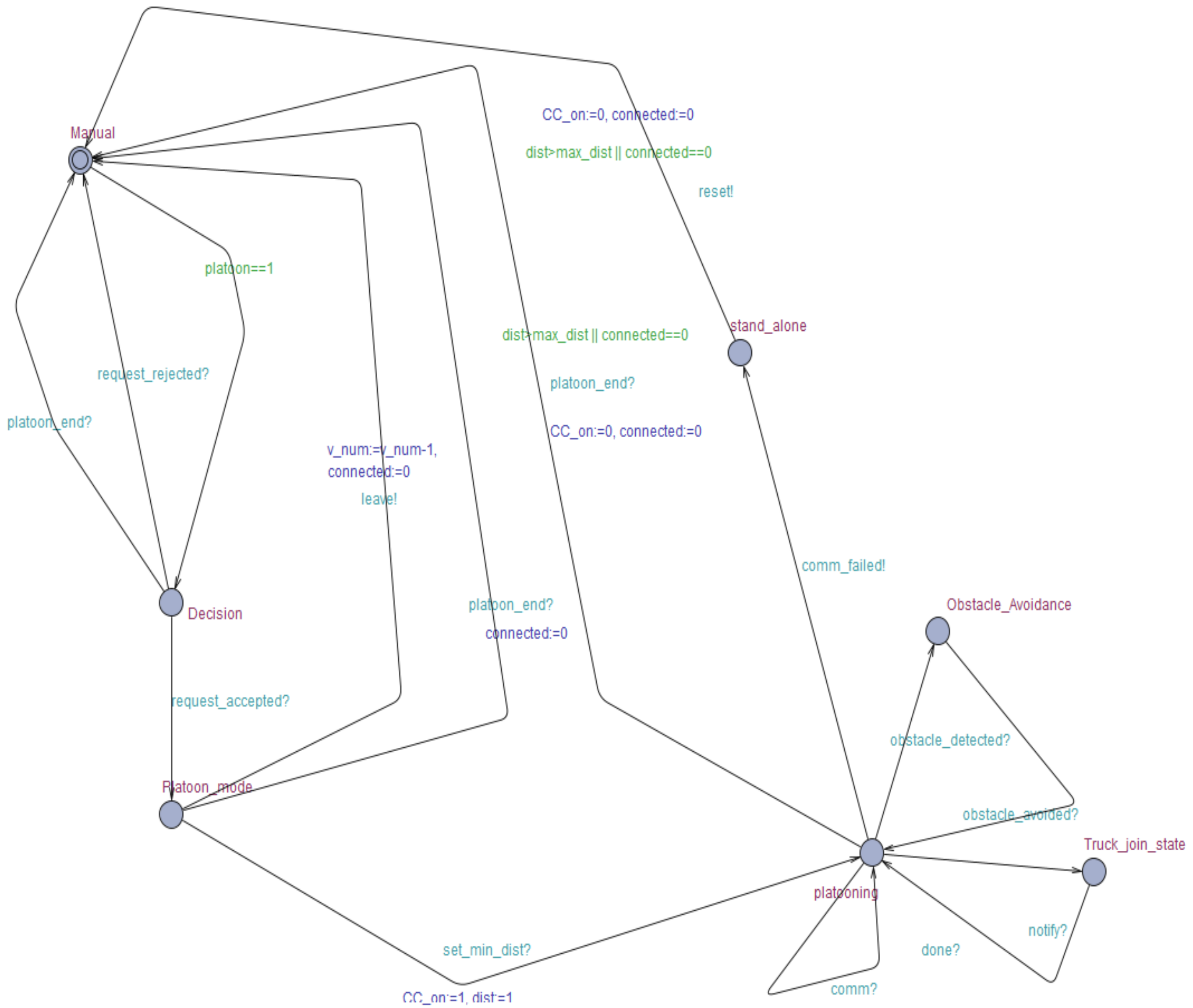
Scenarios Included in the Timed Automata

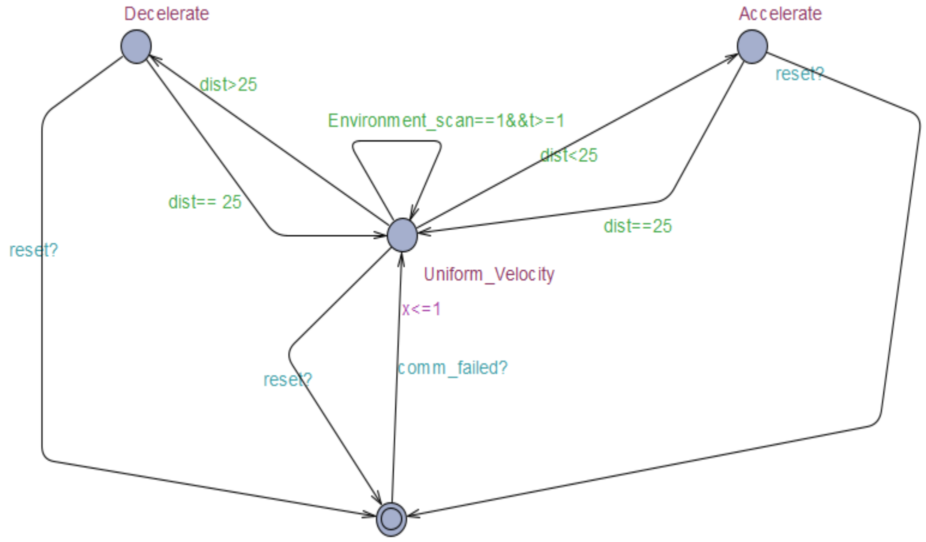
- Maintaining distance
- Communication failure
- Truck Join while platooning
- Obstacle avoidance

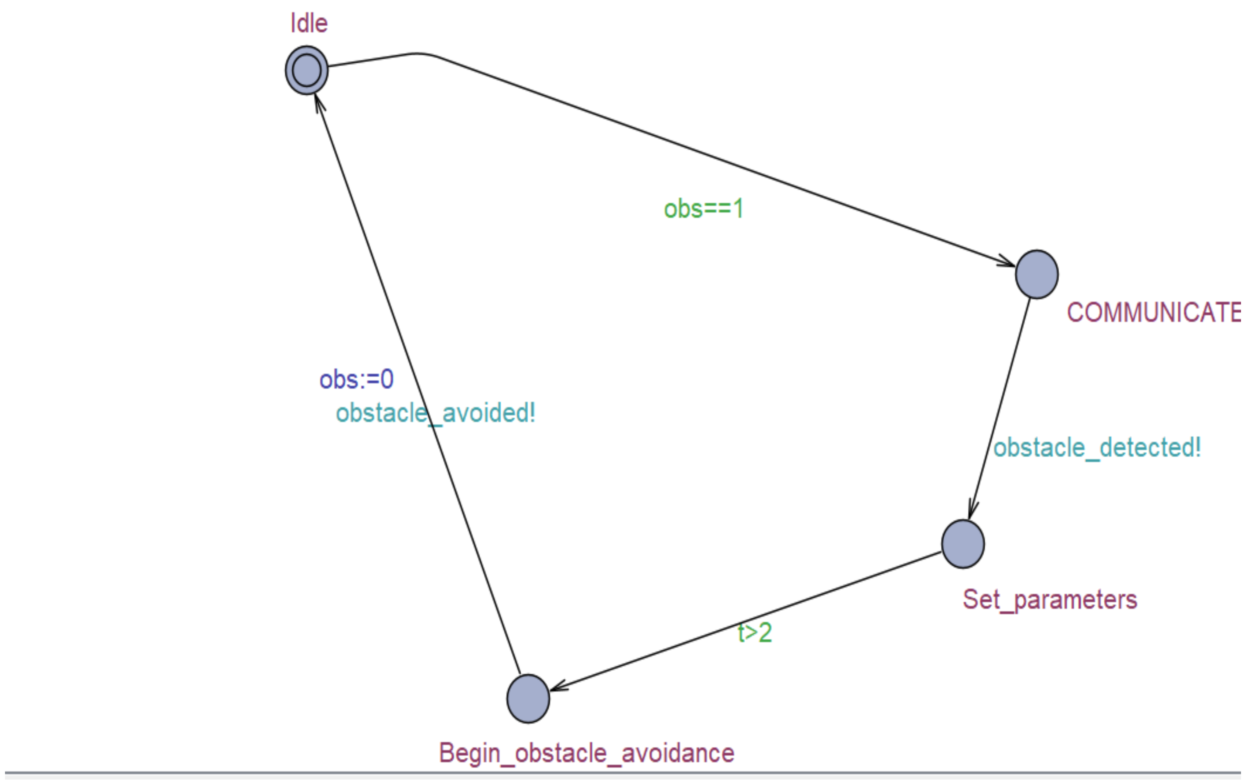
Task 2 and 3

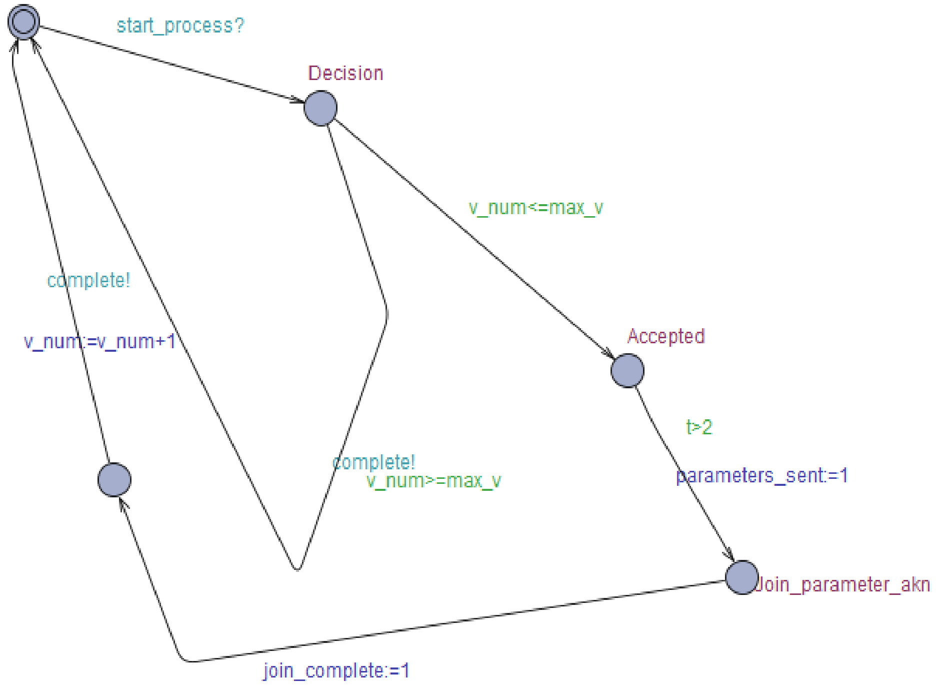


CC_on:=0,
v_num:=v_num-1
connected==0 || dist > max_dist
platoon_end!
Manual
Stand_alone
connected==0 || dist > max_dist
platoon_end!
CC_on:=0,
v_num:=v_num-1
platoon:=1
platoon:=0, v_num:=0
platoon_end!
Accepted
autonomous_on!
Obstacle_Avoidanc
request_accepted!
v_num<=max_v
v_num:=v_num+1
set_min_dist!
CC_on:=1
obstacle_detected?
Begin
obs:=1
Initiate_platoon
Platooning
obstacle_avoided?
request_rejected!
v_num>=max_v
notify!
truck_request<=max_v
truck_request:=truck_request+1
comm!
t>=1
done?
join_request!
Decelerate
Accelerate
Environment_scan==1&&t>=1
dist>25
dist<25
dist:=25
uniform_velocity
x<=1
platoon_end!
platoon_end!
autonomous_on?
platoon_end?
IDLE
platoon_end?
Idle
done!
join_request?
RECONNECT
Decision
complete?
truck_request>=1 && t>3
start_process!
await_process


Manual
CC_on:=0, connected:=0
dist>max_dist || connected==0
reset!
platoon==1
request_rejected?
platoon_end?
v_num:=v_num-1,
connected:=0
leave!
dist>max_dist || connected==0
platoon_end?
CC_on:=0, connected:=0
stand_alone
Decision
request_accepted?
platoon_end?
connected:=0
comm_failed!
Obstacle_Avoidance
obstacle_detected?
obstacle_avoided?
Platoon_mode
Truck_join_state
platooning
notify?
done?
set_min_dist?
comm?
CC_on:=1, dist:=1


Decelerate
Accelerate
dist>25
reset?
Environment_scan==1&&t>=1
dist<25
dist== 25
dist==25
reset?
Uniform_Velocity
x<=1
reset?
comm_failed?


Idle
obs==1
COMMUNICATE
obs:=0
obstacle_avoided!
obstacle_detected!
Set_parameters
t>2
Begin_obstacle_avoidance


start_process?
Decision
v_num<=max_v
complete!
v_num:=v_num+1
Accepted
t>2
complete!
v_num>=max_v
parameters_sent:=1
Join_parameter_akn
join_complete:=1

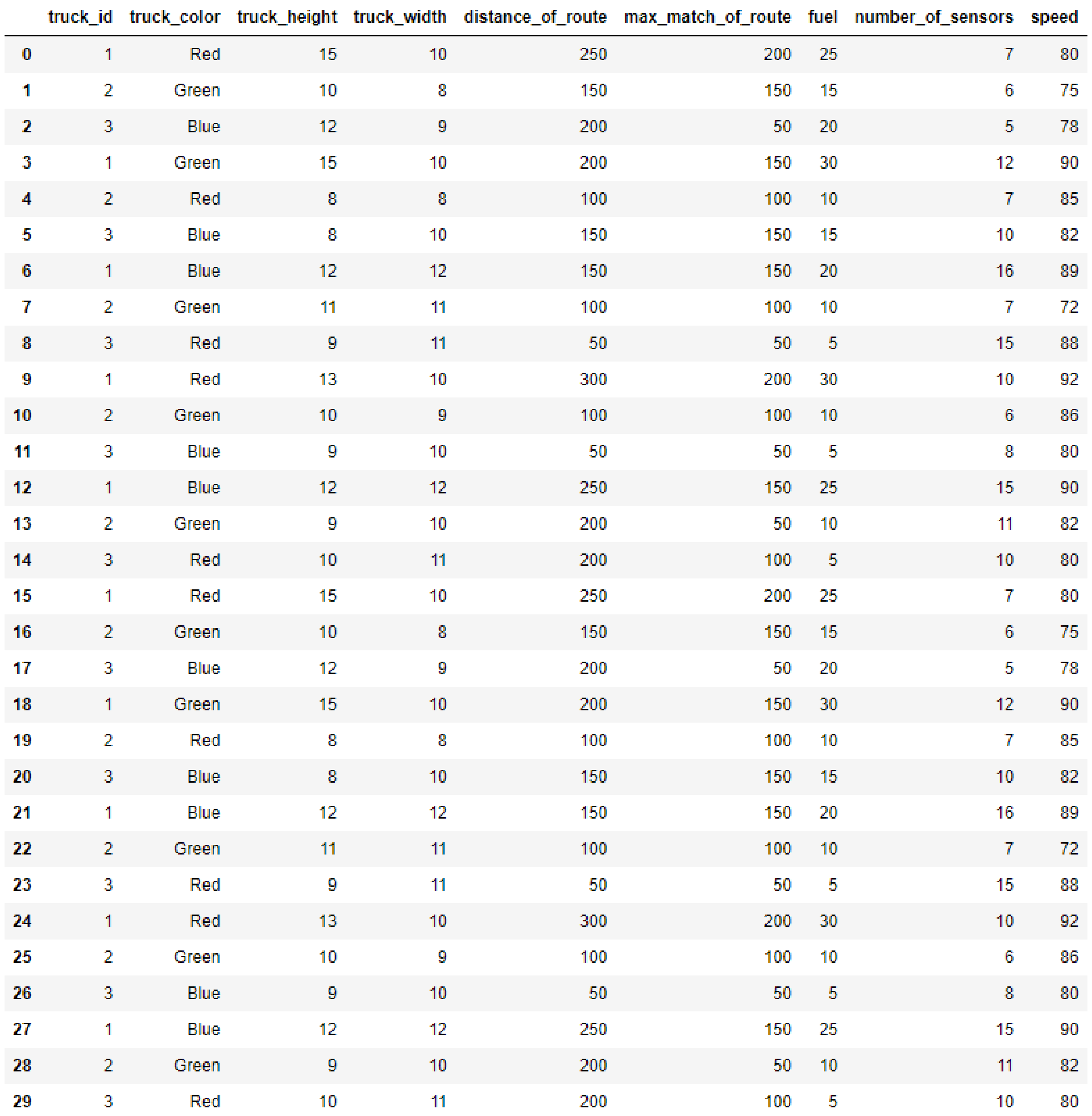
Task 4

- Machine learning algorithm
- Scikit-learn
- Decision Tree

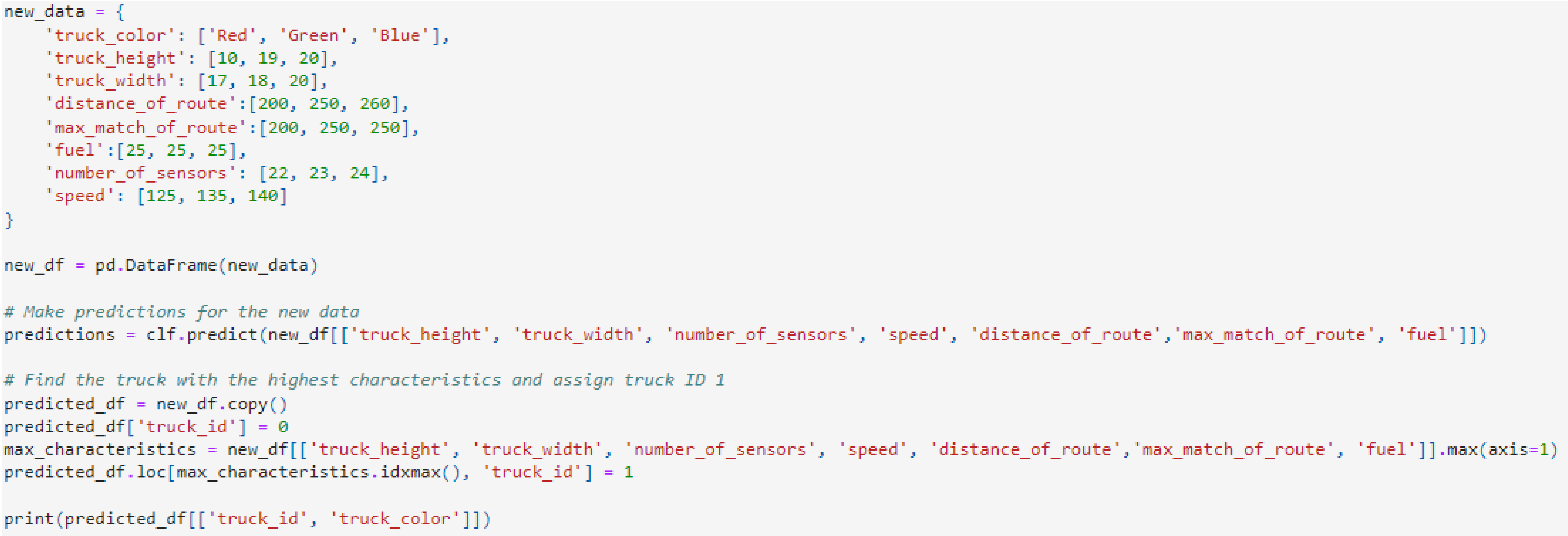
  - Why?

    - Interpretability
    - Feature Importance
    - Data handling

- **Training data set**

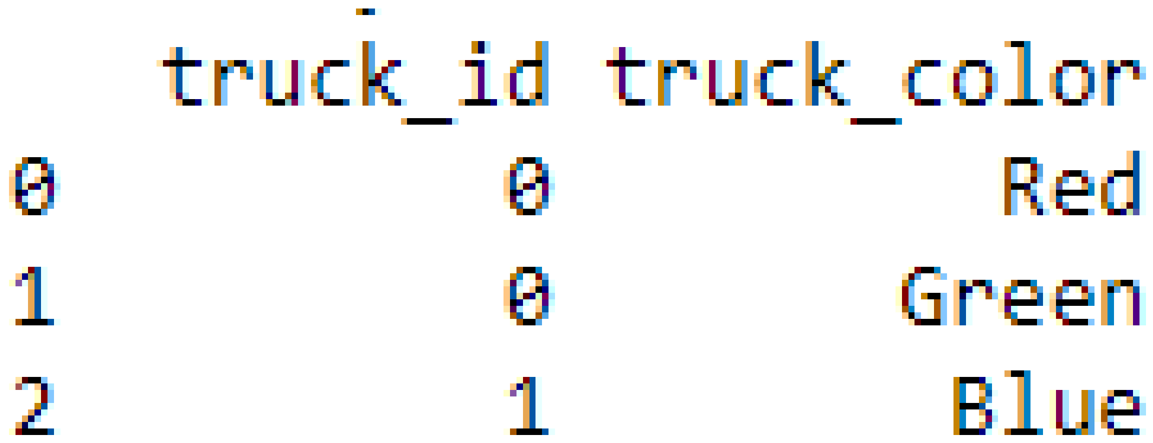


	truck_id	truck_color	truck_height	truck_width	distance_of_route	max_match_of_route	fuel	number_of_sensors	speed
0	1	Red	15	10	250	200	25	7	80
1	2	Green	10	8	150	150	15	6	75
2	3	Blue	12	9	200	50	20	5	78
3	1	Green	15	10	200	150	30	12	90
4	2	Red	8	8	100	100	10	7	85
5	3	Blue	8	10	150	150	15	10	82
6	1	Blue	12	12	150	150	20	16	89
7	2	Green	11	11	100	100	10	7	72
8	3	Red	9	11	50	50	5	15	88
9	1	Red	13	10	300	200	30	10	92
10	2	Green	10	9	100	100	10	6	86
11	3	Blue	9	10	50	50	5	8	80
12	1	Blue	12	12	250	150	25	15	90
13	2	Green	9	10	200	50	10	11	82
14	3	Red	10	11	200	100	5	10	80
15	1	Red	15	10	250	200	25	7	80
16	2	Green	10	8	150	150	15	6	75
17	3	Blue	12	9	200	50	20	5	78
18	1	Green	15	10	200	150	30	12	90
19	2	Red	8	8	100	100	10	7	85
20	3	Blue	8	10	150	150	15	10	82
21	1	Blue	12	12	150	150	20	16	89
22	2	Green	11	11	100	100	10	7	72
23	3	Red	9	11	50	50	5	15	88
24	1	Red	13	10	300	200	30	10	92
25	2	Green	10	9	100	100	10	6	86
26	3	Blue	9	10	50	50	5	8	80
27	1	Blue	12	12	250	150	25	15	90
28	2	Green	9	10	200	50	10	11	82
29	3	Red	10	11	200	100	5	10	80



```
new_data = {
    'truck_color': ['Red', 'Green', 'Blue'],
    'truck_height': [10, 19, 20],
    'truck_width': [17, 18, 20],
    'distance_of_route':[200, 250, 260],
    'max_match_of_route':[200, 250, 250],
    'fuel':[25, 25, 25],
    'number_of_sensors': [22, 23, 24],
    'speed': [125, 135, 140]
}

new_df = pd.DataFrame(new_data)

# Make predictions for the new data
predictions = clf.predict(new_df[['truck_height', 'truck_width', 'number_of_sensors', 'speed', 'distance_of_route','max_match_of_route', 'fuel']])

# Find the truck with the highest characteristics and assign truck ID 1
predicted_df = new_df.copy()
predicted_df['truck_id'] = 0
max_characteristics = new_df[['truck_height', 'truck_width', 'number_of_sensors', 'speed', 'distance_of_route','max_match_of_route', 'fuel']].max(axis=1)
predicted_df.loc[max_characteristics.idxmax(), 'truck_id'] = 1

print(predicted_df[['truck_id', 'truck_color']])
```

- **Lead truck prediction**



```
   truck_id truck_color
0         0         Red
1         0       Green
2         1        Blue
```

Task 5

- Joining truck with lead truck integration
- Maintaining distance with lead truck integration

Thank you for your attention!