Truck Platooning

Autonomous Systems
A Lab

- **(Group B)**
- **Jasmeet Singh Matta**
- **Moshiour Rahman Prince**
- **Shihab Ud Doula**

Introduction

- **Truck platooning:** This technique involves multiple trucks driving in close formation, coordinated through communication systems.

- **Project Goal:** Model, simulate and validate truck platooning scenarios

Figure 1: Truck Platooning Concept

Motivation & Objective

Motivation:

- Safety: Reduces accidents.
- Efficiency: Lowers fuel consumption.
- Traffic Flow: Optimizes road usage.

Objectives:

- Model critical driving scenarios.
- Ensure system robustness and stability.
- Integrate advanced modelling and real-time data processing.

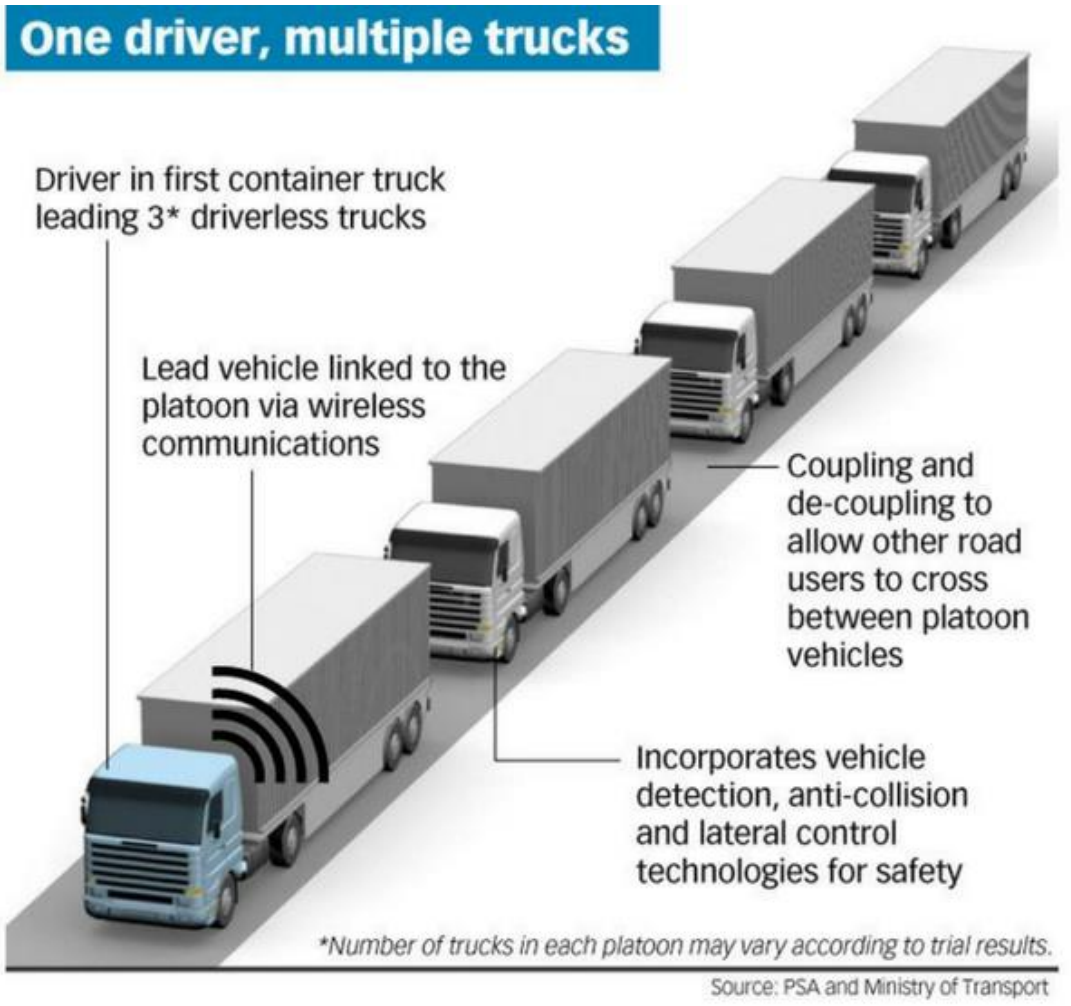


Figure 2: How does truck platooning work?

Identifying Key Scenarios

1

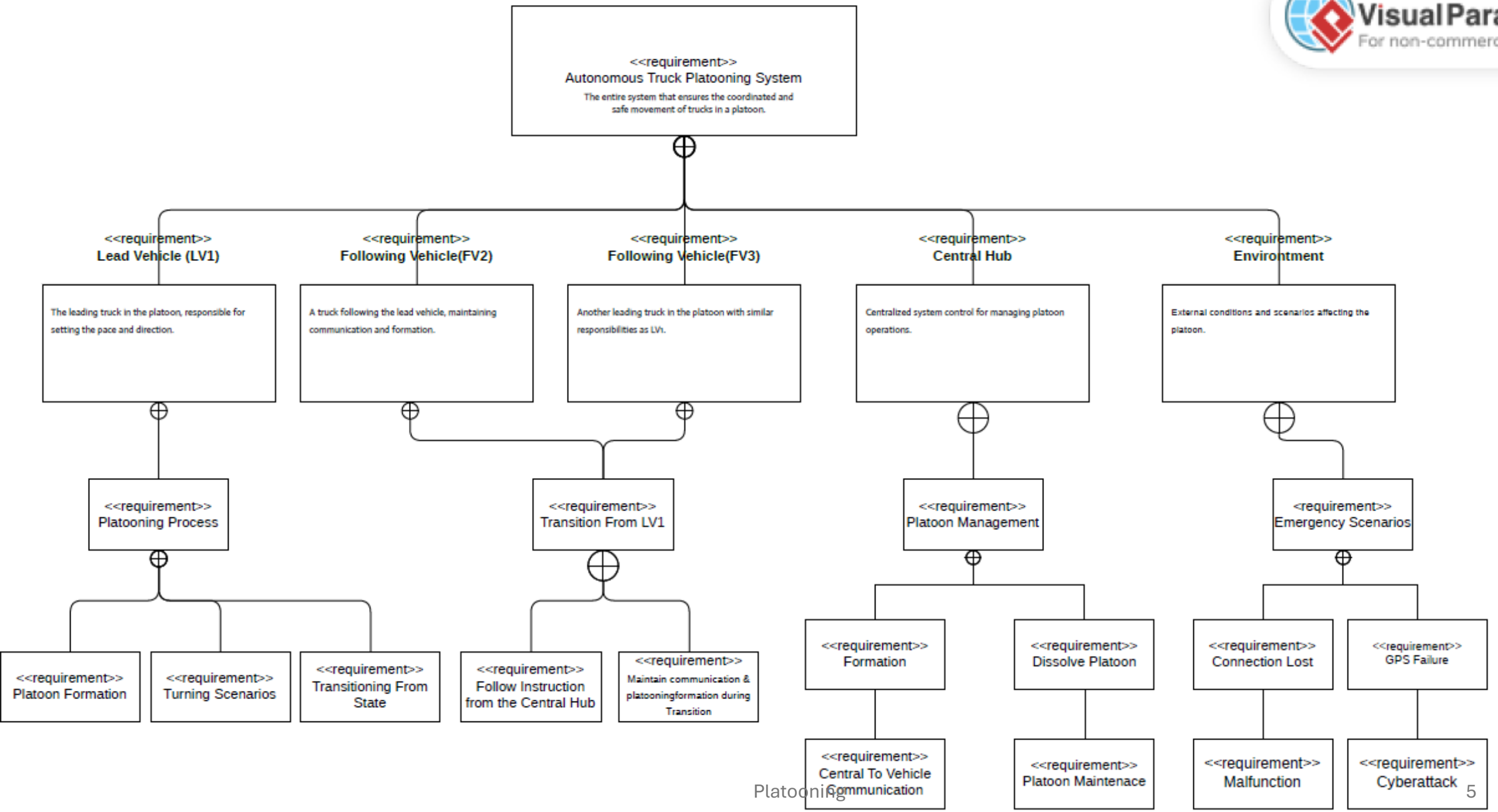
Obstacle Avoidance: Detect and respond to obstacles.

2

Lane Changing and Steering: Manage lane changes and steering to prevent collisions.

3

Environmental Model: Simulate system behavior in various environmental conditions.



- **Environmental Model:** Response to lost connections and emergencies.

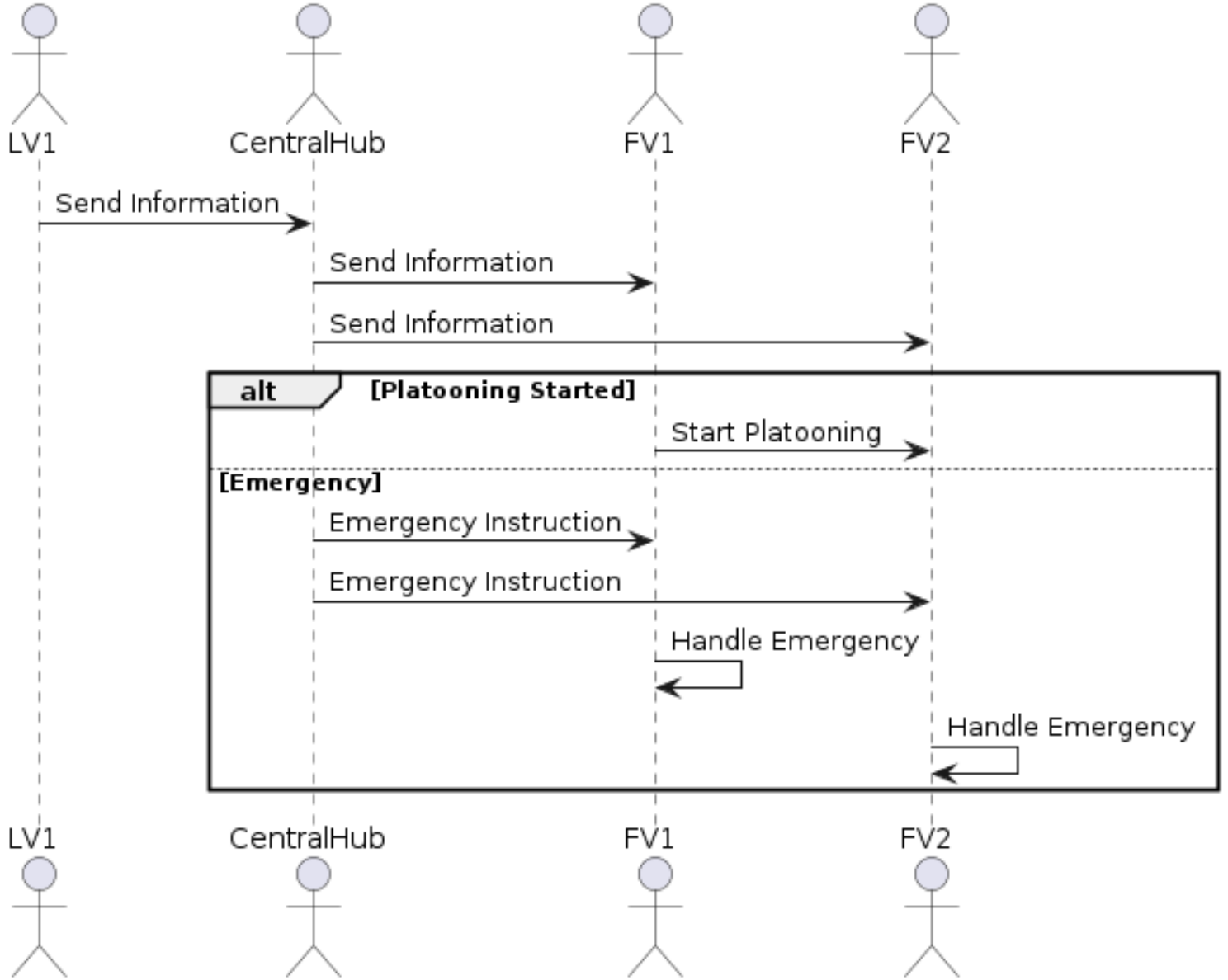


Figure 5: Emergency Handleing

- **Lane Changing:** Process for lane change requests and system responses.

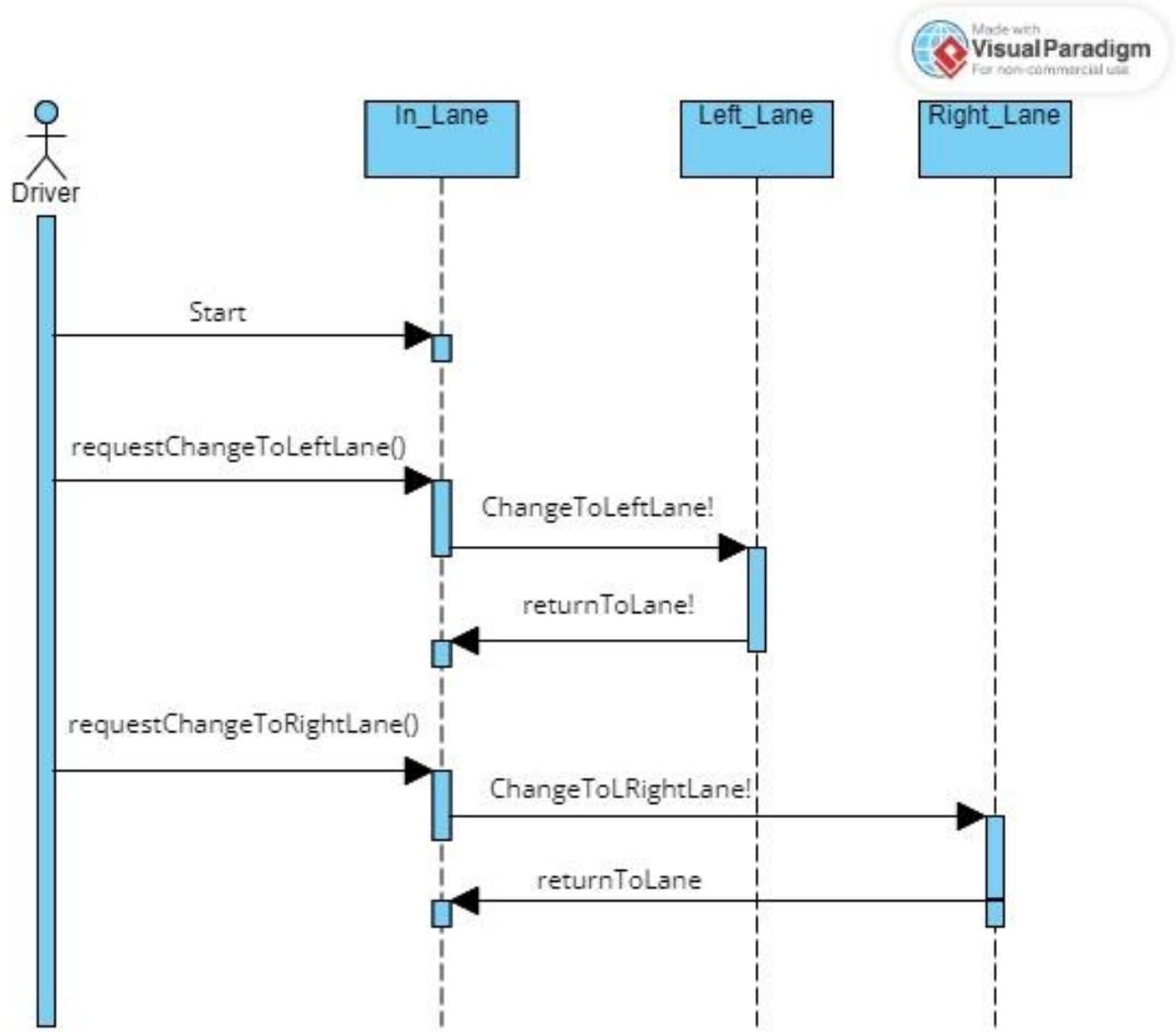
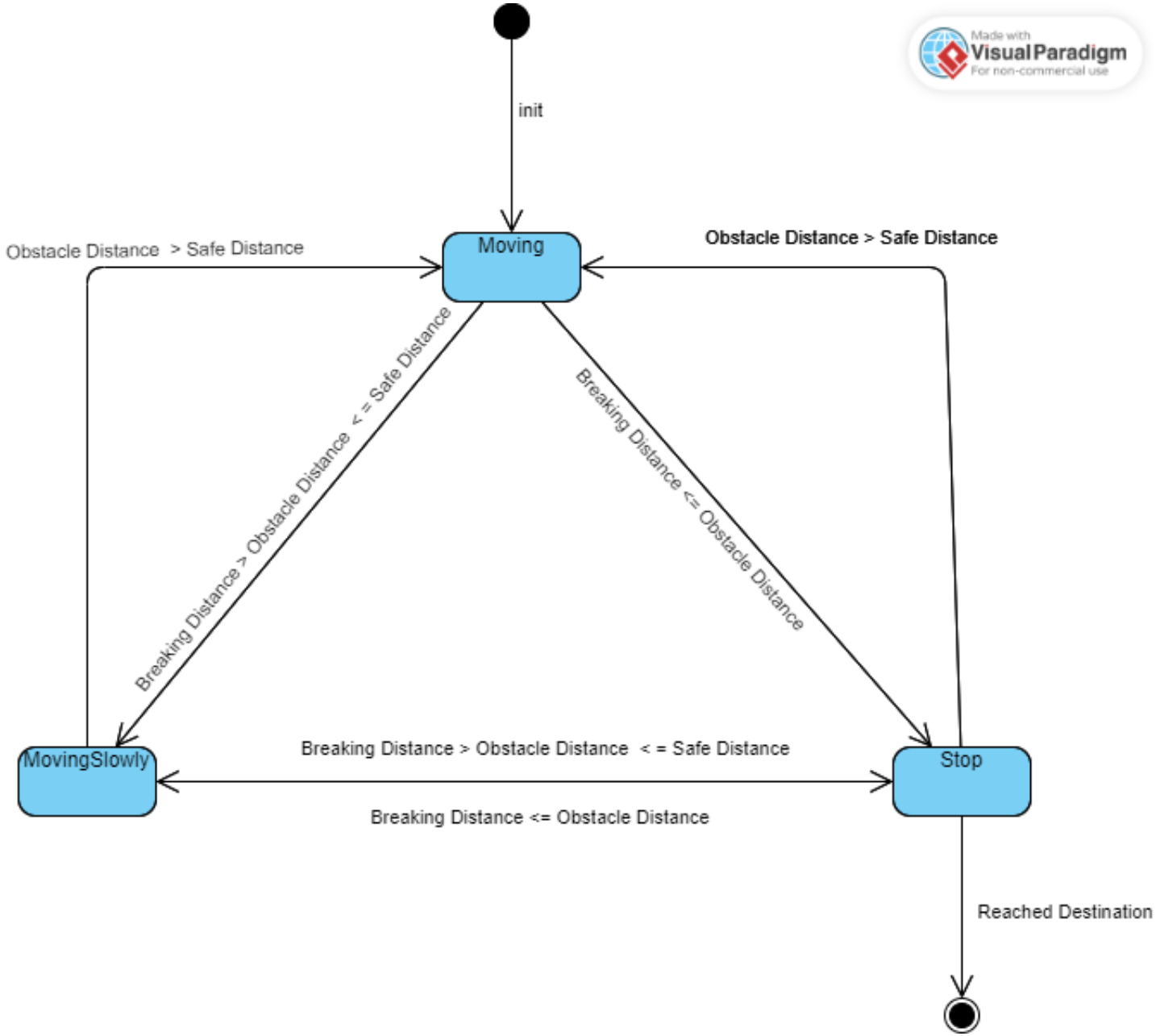


Figure 5: Lane Changing & Steering

- **Obstacle Avoidance:** State machine diagram for avoiding obstacle



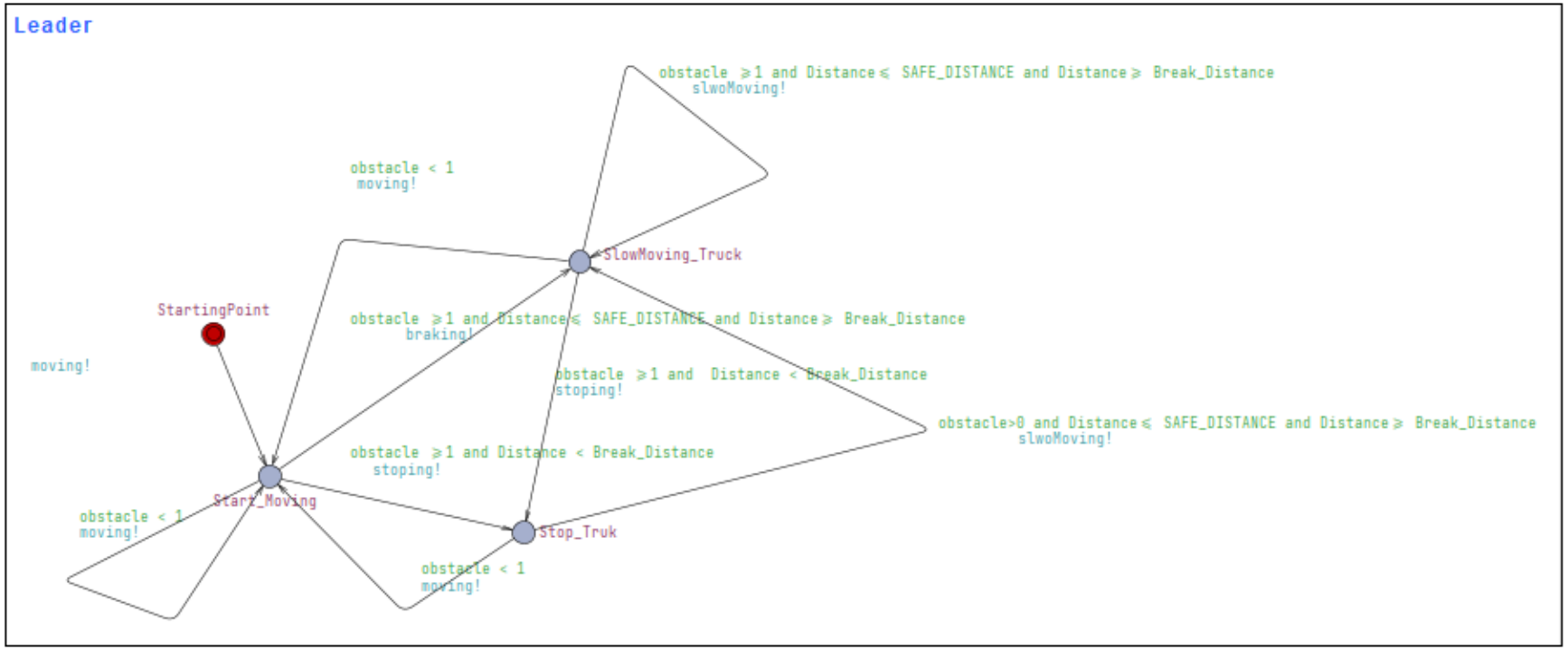
Modeling Scenarios with UPPAAL

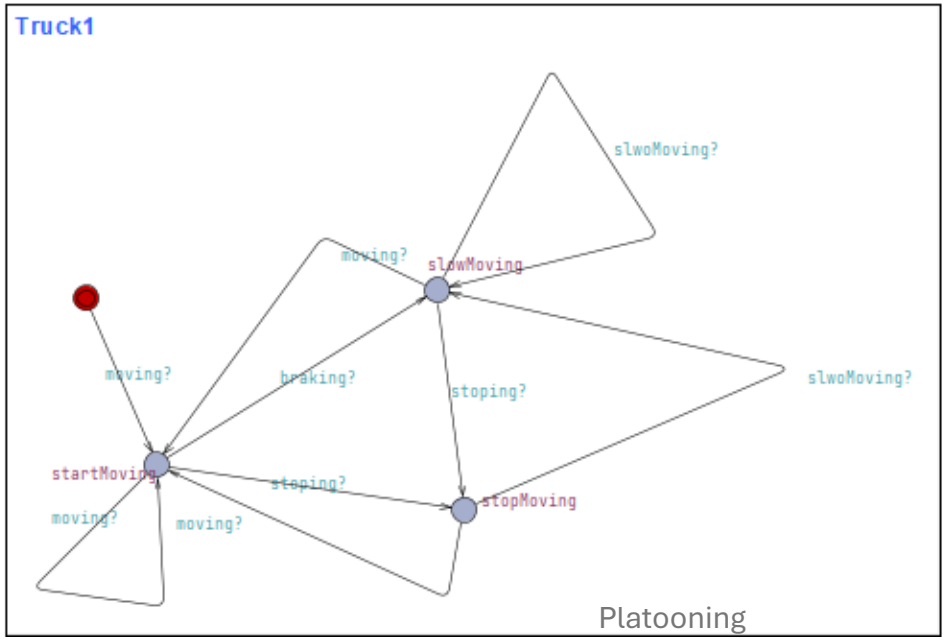
Obstacle Avoidance: Truck detects and responds to obstacles.

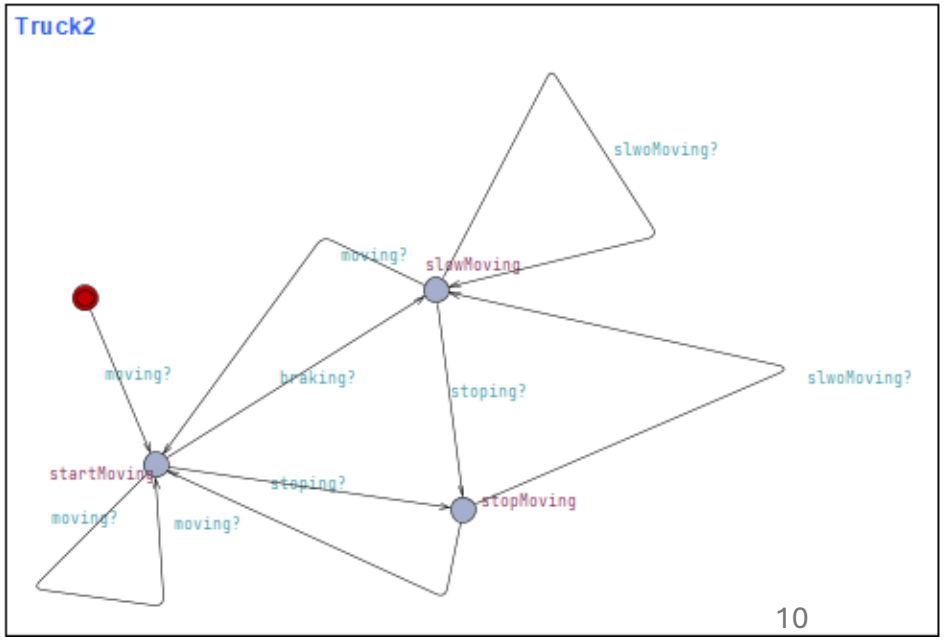
Lane Changing: Checks lane availability and executes lane changes safely.

System Integration: Combines all scenarios into one cohesive model, ensuring dynamic responses.

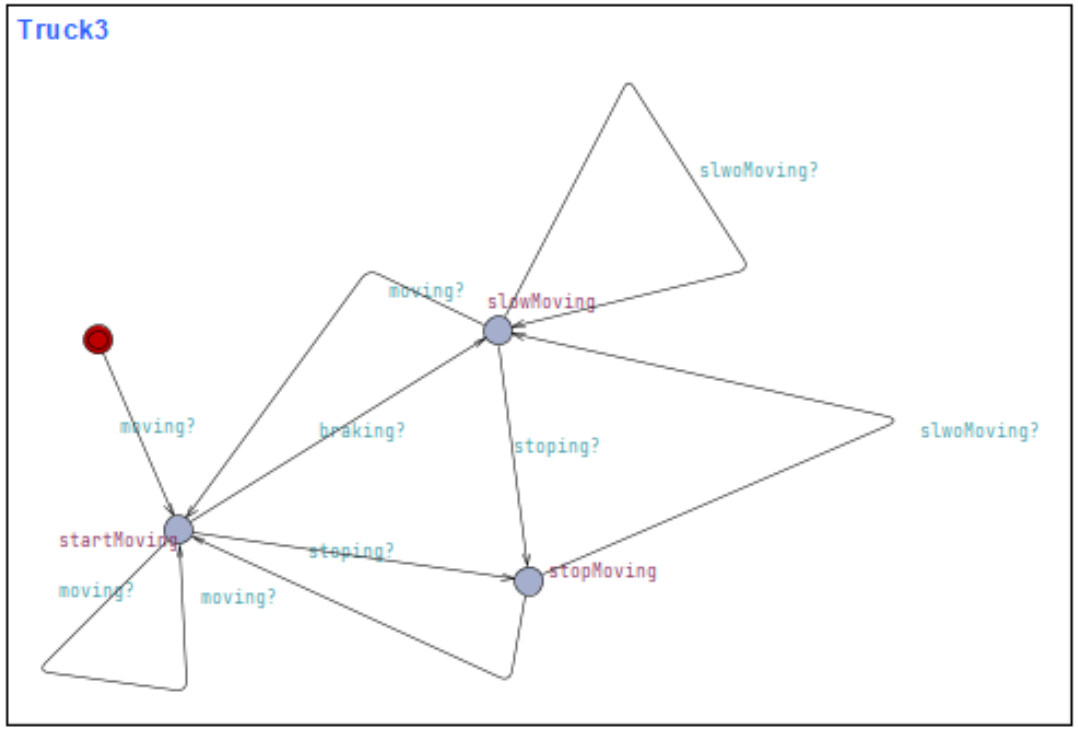
Obstacle Avoidance

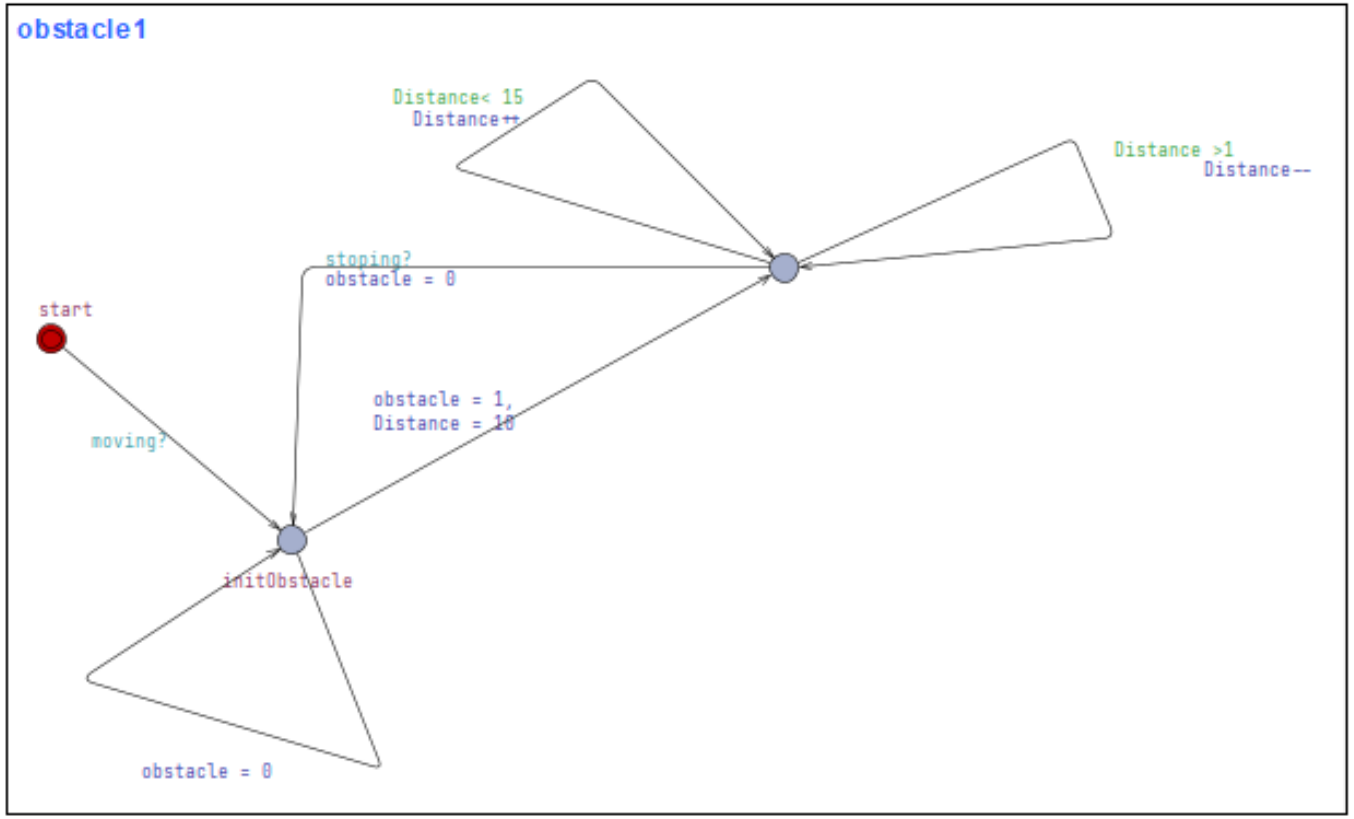




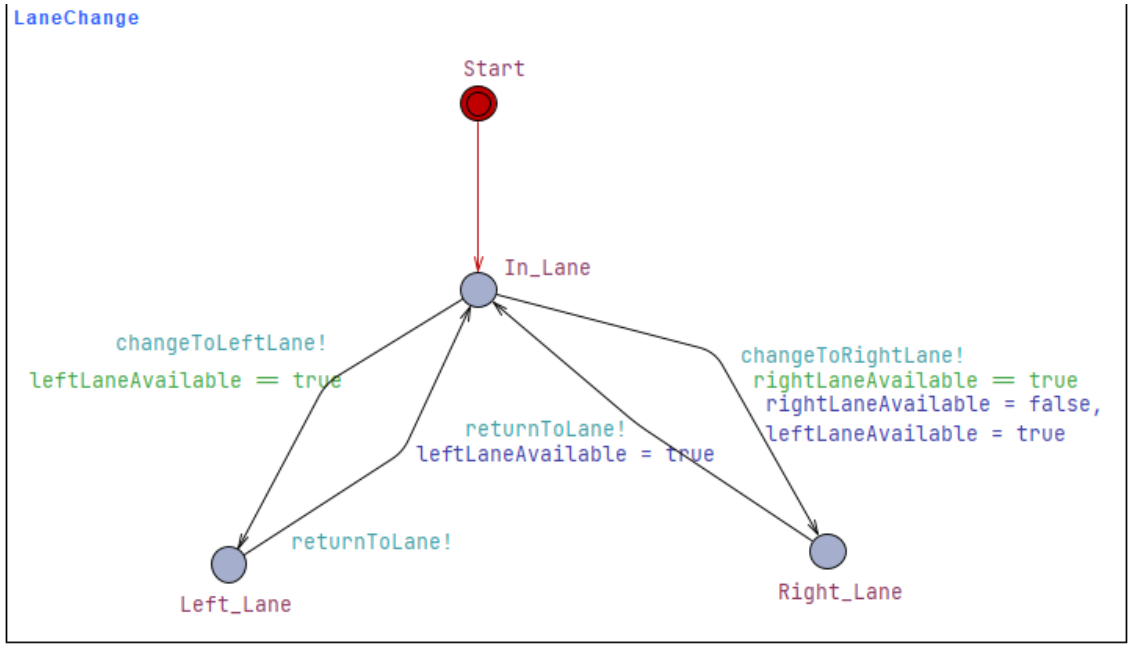


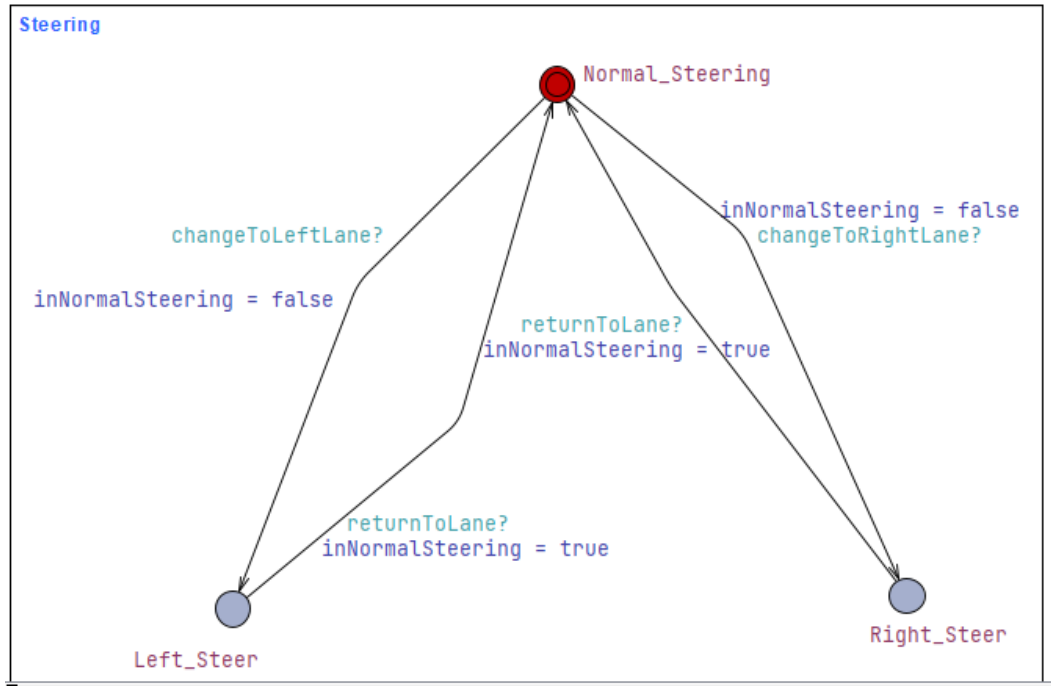
Obstacle Avoidance



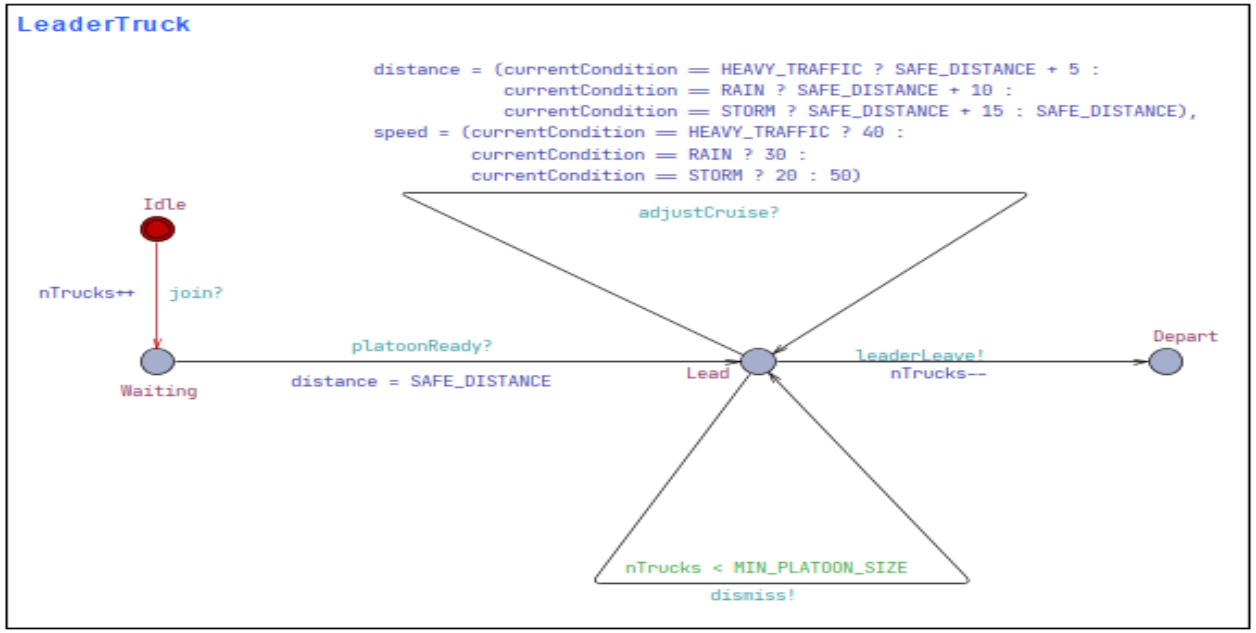


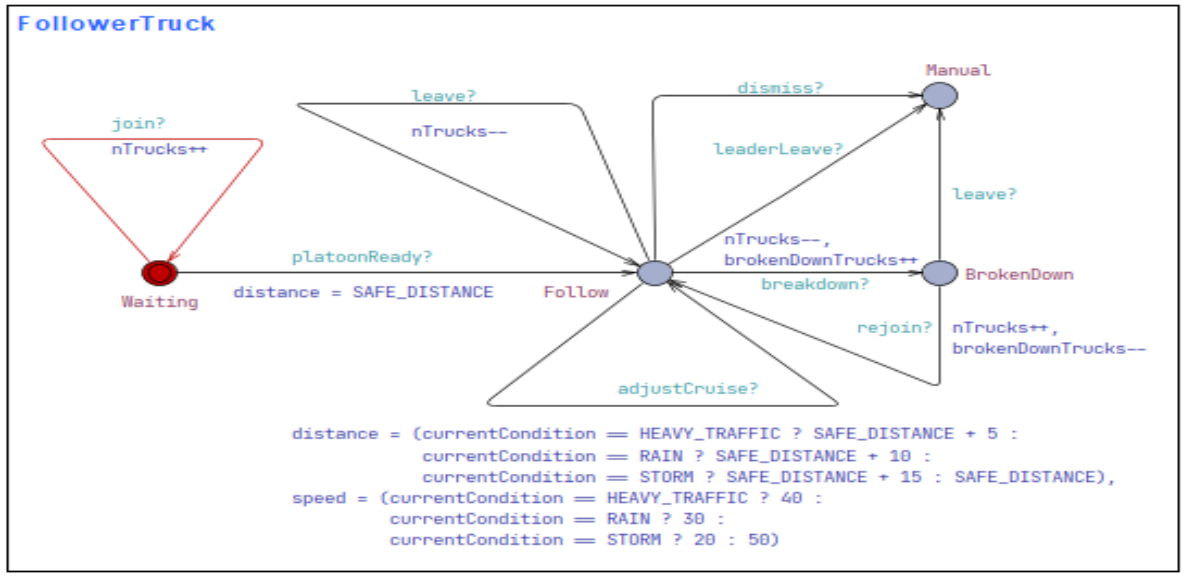
Lane Changing & Steering

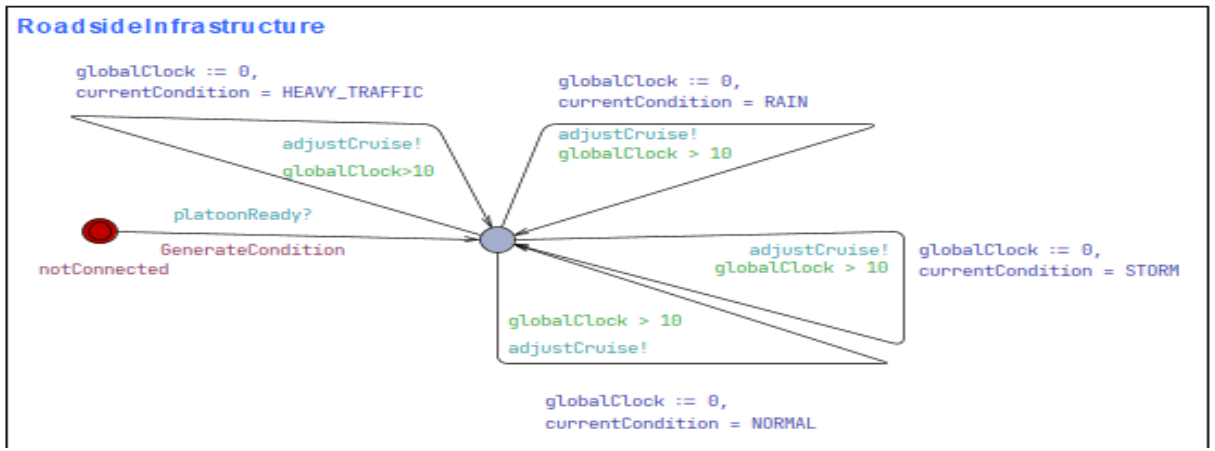


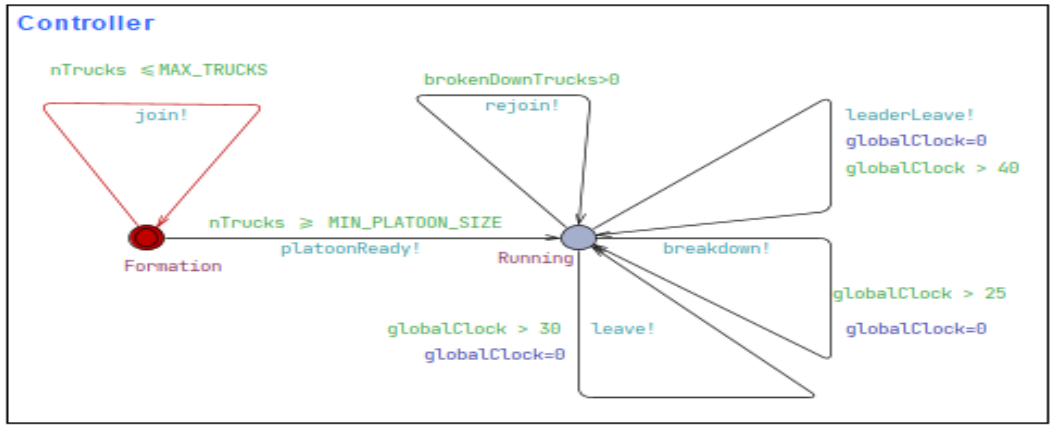


Environmental Model







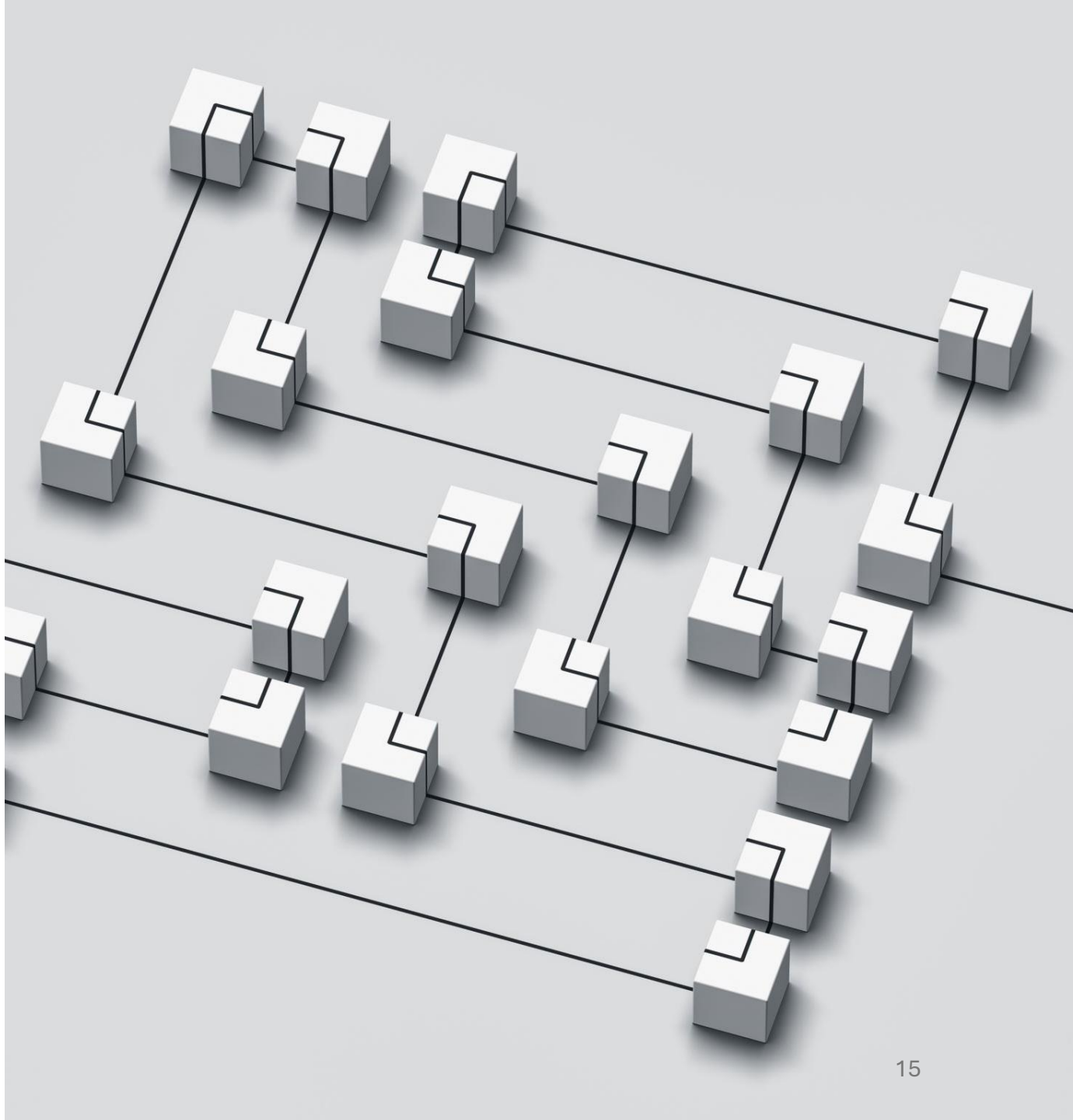


Combined Model

- Full video available tutorial in main presentation pptx file.
- Combined model with code and each scenario also available on https://github.com/JasmeetMatta/TruckPlatooning/tree/main

Decision Making with Machine Learning

- **Leader Vehicle Determination**
- Implemented a logistic regression using Scikit-learn.
- **Key Parameters:** Years of experience, Aerodynamic Score, Technological score
- **Training:** Splitting a dataset into training and testing data, creating a logistic regression model, and then fitting that model to the training data.



Machine Learning Implementation

years_of_experience	aerodynamics_score	technological_score
4	45	10
6	29	7
8	34	9
5	50	9
4	19	8
7	12	10
20	29	2
5	9	8
16	10	5
11	23	8
11	7	10
2	45	4
14	25	4
1	38	3
5	17	5
8	23	10
12	11	7
7	6	7
17	41	9
9	33	2
6	49	5
0	6	5
18	20	6
20	34	7

Parameters

- years_of_experience = (0, 20)
- aerodynamics_score = (5, 50)
- technological_score = (1, 10)
- 5000 samples

Data Preparation

years_of_experience	aerodynamics_score	technological_score	Leader	Leadership_score
4	45	10	1	540
6	29	7	0	369
8	34	9	0	443
5	50	9	1	588
4	19	8	0	266
7	12	10	0	225
20	29	2	0	404
5	9	8	0	171
16	10	5	0	215
11	23	8	0	341
11	7	10	0	195
2	45	4	1	488
14	25	4	0	348
1	38	3	0	406
5	17	5	0	230
8	23	10	0	340
12	11	7	0	219
7	6	7	0	144
17	41	9	1	558
9	33	2	1	389

Score = $w1 \cdot A + w2 \cdot B + w3 \cdot C$

- $w1$= 5; $w2$ = 10; $w3$ = 7;
  - A= years of experience
  - B = aerodynamics score
  - C = technological score

- If score > 450 : leader = True

Model Training

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X,Y,train_size=0.8)
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
```

```
model.fit(X_train, Y_train)
```

Leader Selection

```
test = pd.read_csv("test.csv")
test.head()
```

	years_of_experience	aerodynamics_score	technological_score
0	4	45	10
1	7	30	5
2	5	20	7

```
model.predict(test)
```

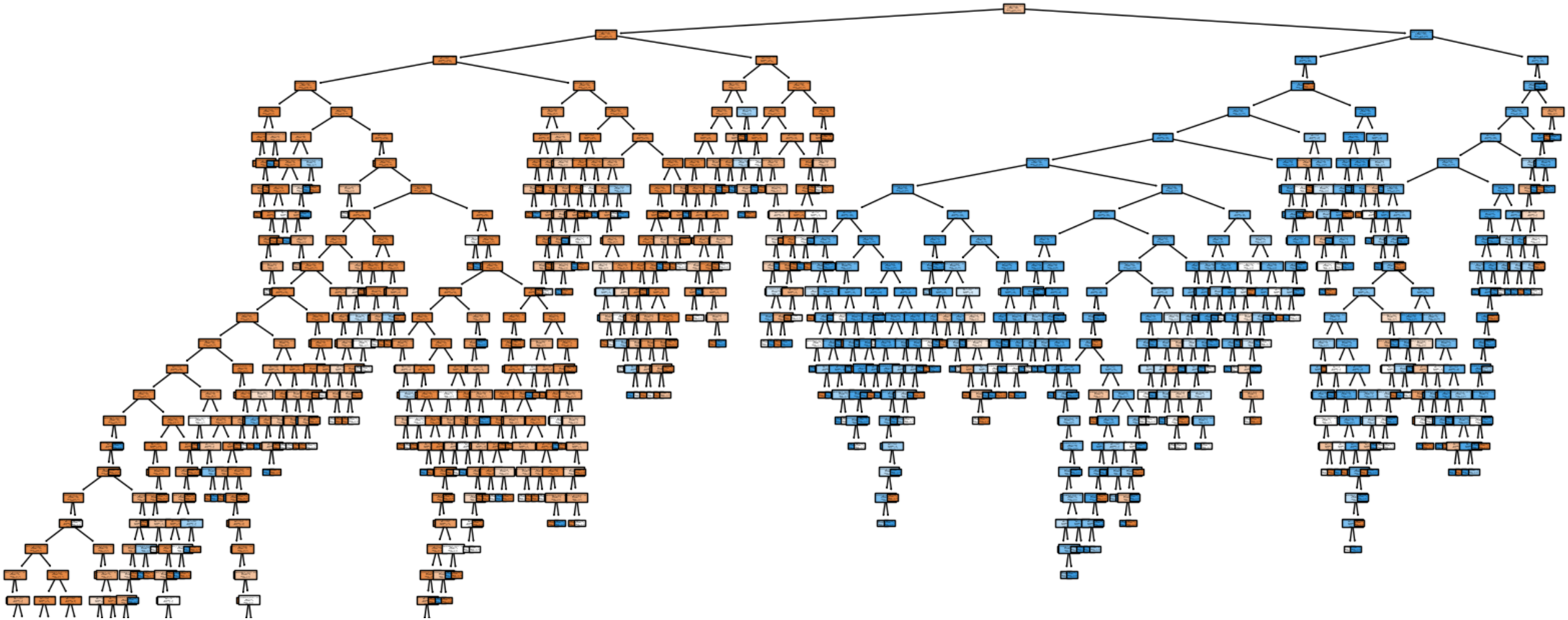
```
array([1, 0, 0], dtype=int64)
```

Decision Tree for Leader Vehicle Determination

Accuracy: 87%

Actual \ Predicted	**0 (No Leader)**	**1 (Leader)**
0 (No Leader)	642	58
1(Leader)	70	230

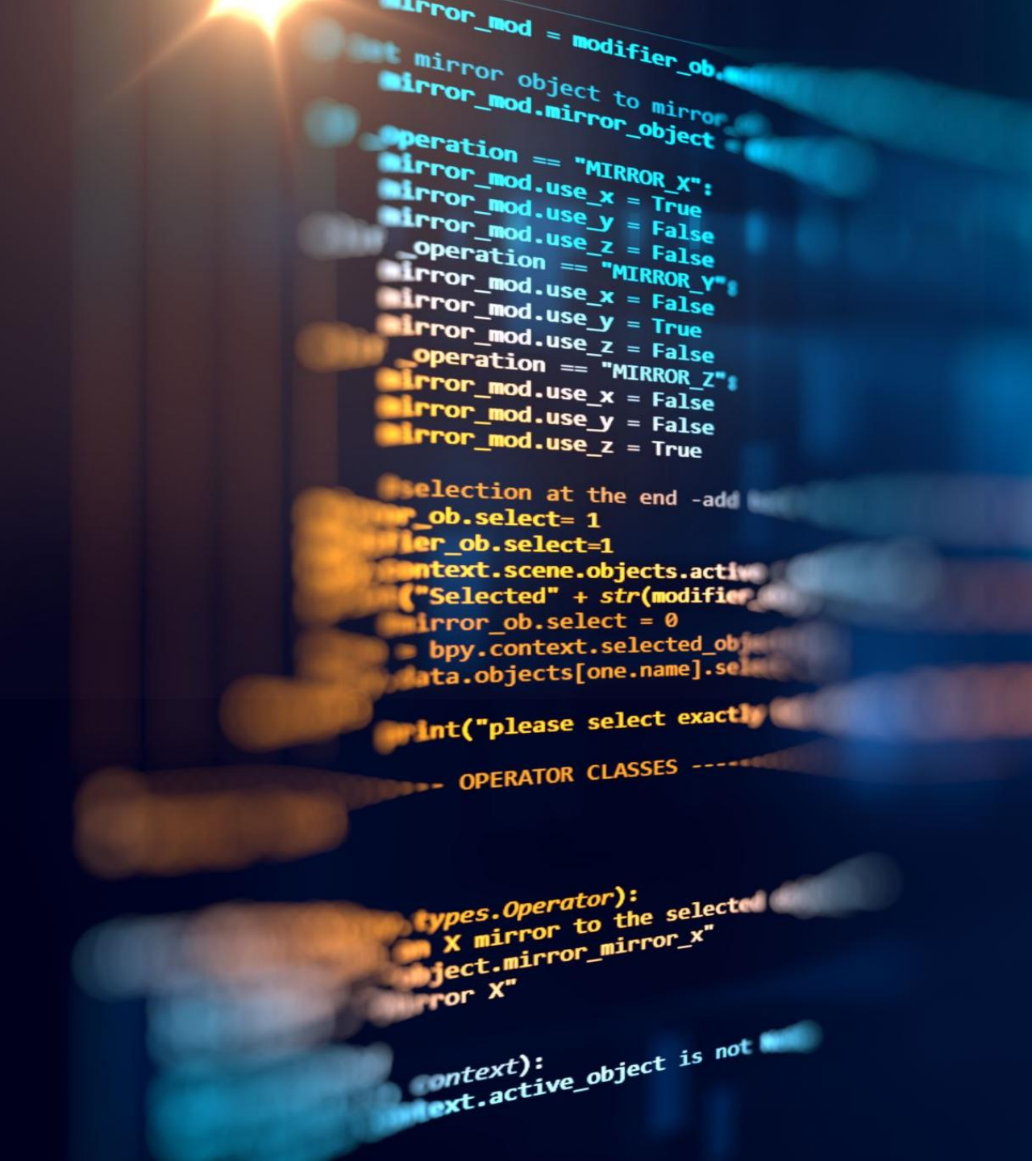
```
# Train a Decision Tree model
model = DecisionTreeClassifier(random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy:.2f}')
print('Classification Report:')
print(classification_report(y_test, y_pred))
print('Confusion Matrix:')
print(confusion_matrix(y_test, y_pred))
```



Analysing Solutions

```
score = Linearmodel.score(X_test,x_Linearpred)
print("Linear Regression",score*100,'%')
print("Logistic Regression Accuracy:", accuracy_score(Y_test, x_Logisticpred)*100,'%')
print("DecisionTreeClassifier Accuracy:", accuracy_score(Y_test, x_DTCpred)*100,'%')
```

```
Linear Regression 45.977159834190054 %
Logistic Regression Accuracy: 86.7 %
DecisionTreeClassifier Accuracy: 93.5 %
```



Simulation Environment (Python) need update

- **Development**: Built a simulation environment to test scenarios and the learning algorithm.
- **Integration**: Ensured synchronization of scenarios and the learning algorithm.
- **Tools**: Implemented in Python for real-time data processing.

Lane Changing

```
class TruckPlatooning:
    def __init__(self):
        self.leftLaneAvailable = True
        self.rightLaneAvailable = True
        self.currentLane = 'Start'

    def change_to_left_lane(self):
        if self.leftLaneAvailable:
            self.currentLane = 'Left_Lane'
            print("Changed to left lane.")
            self.rightLaneAvailable = True

    def change_to_right_lane(self):
        if self.rightLaneAvailable:
            self.currentLane = 'Right_Lane'
            print("Changed to right lane.")
            self.leftLaneAvailable = True

    def return_to_lane(self):
        self.currentLane = 'In_Lane'
        print("Returned to main lane.")
```

Simulation Overview:

- Demonstrates lane changing based on lane availability

```
Current Lane: Start
Changed to right lane.
Returned to main lane.
Current Lane: In_Lane
Changed to left lane.
Returned to main lane.
Current Lane: In_Lane
Changed to left lane.
Returned to main lane.
Current Lane: In_Lane
Changed to right lane.
Returned to main lane.
Current Lane: In_Lane
Changed to right lane.
Returned to main lane.
Current Lane: In_Lane
Changed to right lane.
Returned to main lane.
Current Lane: In_Lane
Changed to left lane.
Returned to main lane.
Current Lane: In_Lane
Changed to right lane.
Returned to main lane.
Current Lane: In_Lane
Changed to right lane.
Returned to main lane.
```

Environmental Changes

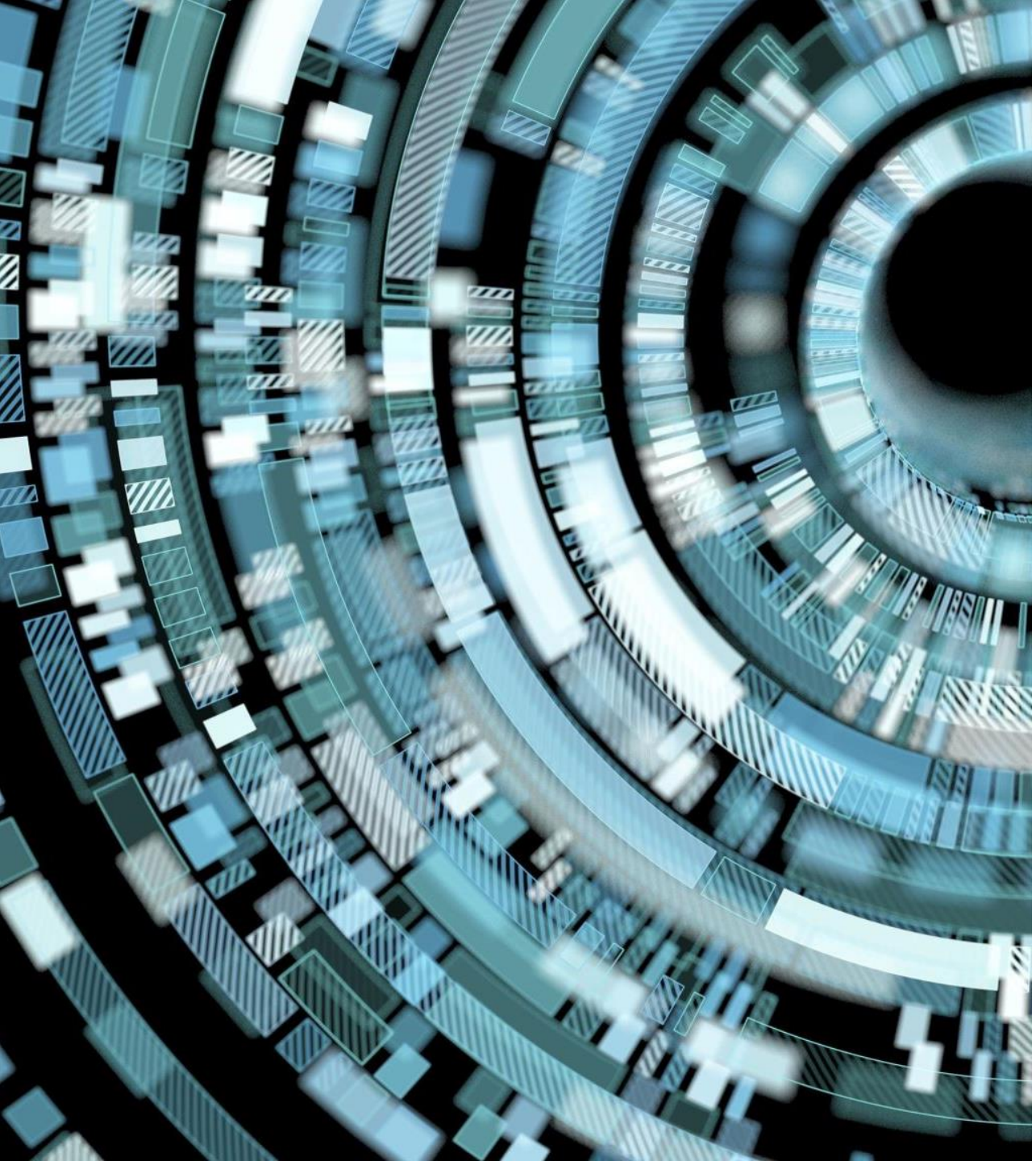
```
Truck joined. Total trucks: 1
Adjusting cruise becuase of NORMAL: New distance 10, New speed 50
Adjusting cruise becuase of HEAVY_TRAFFIC: New distance 15, New speed 40
Truck joined. Total trucks: 2
Adjusting cruise becuase of RAIN: New distance 20, New speed 30
Truck joined. Total trucks: 3
Adjusting cruise becuase of HEAVY_TRAFFIC: New distance 15, New speed 40
Platoon is now ready!
Adjusting cruise becuase of STORM: New distance 25, New speed 20
Truck joined. Total trucks: 4
Adjusting cruise becuase of NORMAL: New distance 10, New speed 50
Truck joined. Total trucks: 5
Adjusting cruise becuase of HEAVY_TRAFFIC: New distance 15, New speed 40
Adjusting cruise becuase of RAIN: New distance 20, New speed 30
```

Obstacle Avoidance

```
Leader: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 1: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 2: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 3: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Leader: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 1: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 2: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 3: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Leader: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 1: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 2: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 3: Speed=stop, Pedal=hardBrake, BreakDistance=5
Leader: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 1: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 2: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 3: Speed=stop, Pedal=hardBrake, BreakDistance=5
Leader: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 1: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 2: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 3: Speed=Max, Pedal=noBrake, BreakDistance=0
Leader: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 1: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 2: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 3: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Leader: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 1: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 2: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 3: Speed=stop, Pedal=hardBrake, BreakDistance=5
Leader: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 1: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 2: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 3: Speed=Max, Pedal=noBrake, BreakDistance=0
Leader: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 1: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 2: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 3: Speed=Max, Pedal=noBrake, BreakDistance=0
Leader: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 1: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 2: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 3: Speed=stop, Pedal=hardBrake, BreakDistance=5
```

Overall Simulation

```
Current Lane: Start
Changed to left lane.
Returned to main lane.
Leader: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 1: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 2: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 3: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Adjusting cruise becuase of HEAVY_TRAFFIC: New distance 15, New speed 40
Leader: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 1: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 2: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Follower 3: Speed=slow - Same speed as vehicle ahead, Pedal=lightBrake, BreakDistance=0
Adjusting cruise becuase of RAIN: New distance 20, New speed 30
Leader: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 1: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 2: Speed=Max, Pedal=noBrake, BreakDistance=0
Follower 3: Speed=Max, Pedal=noBrake, BreakDistance=0
Current Lane: In_Lane
Changed to left lane.
Returned to main lane.
Adjusting cruise becuase of RAIN: New distance 20, New speed 30
Adjusting cruise becuase of STORM: New distance 25, New speed 20
Current Lane: In_Lane
Changed to left lane.
Returned to main lane.
Adjusting cruise becuase of STORM: New distance 25, New speed 20
Current Lane: In_Lane
Changed to left lane.
Returned to main lane.
Leader: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 1: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 2: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 3: Speed=stop, Pedal=hardBrake, BreakDistance=5
Leader: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 1: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 2: Speed=stop, Pedal=hardBrake, BreakDistance=5
Follower 3: Speed=stop, Pedal=hardBrake, BreakDistance=5
Current Lane: In_Lane
Changed to left lane.
Returned to main lane.
```



Conclusion and Future Work

- Modeled and simulated key truck platooning scenarios
- Demonstrated integration of UML, UPPAAL, and Python.

- **Future Work:** Potential improvements include enhancing the learning algorithm and expanding the simulation environment.

References

- Milford, M. (2017, November 20). Coming soon to a highway near you—truck platooning. The Conversation. https://phys.org/news/2017-11-highway-youtruck-platooning.html
- Truck platooning. (n.d.). Rapp AG. https://www.rapp.ch/de/stories/truck-platooning
- Designing and Scheduling Cost-Efficient Tours by Using the Concept of Truck Platooning - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Basic-mechanisms-behind-the-concept-of-truck-platooning-based-on-Alam-2014-and-Janssen_fig1_338410373 [accessed 4 Jul, 2024]
- Deng, Q. (2016). A General Simulation Framework for Modeling and Analysis of Heavy-Duty Vehicle Platooning. IEEE Transactions on Intelligent Transportation Systems, 17(11), 3252–3262. https://doi.org/10.1109/tits.2016.2548502
- Duan, D. (2017, November 23). Truck Platooning: the band of Semi-Trailers. Labroots. https://www.labroots.com/trending/chemistry-and-physics/7405/band-semi-trailers-truck-platooning


Questions?