

MathWorks
**AUTOMOTIVE
CONFERENCE 2024**
Europe

Automated Driving in the Urban Environment

Advait Valluri, MathWorks



*Seo-Wook Park, MathWorks
contributing author*



What will we learn today regarding virtual validation of AD?

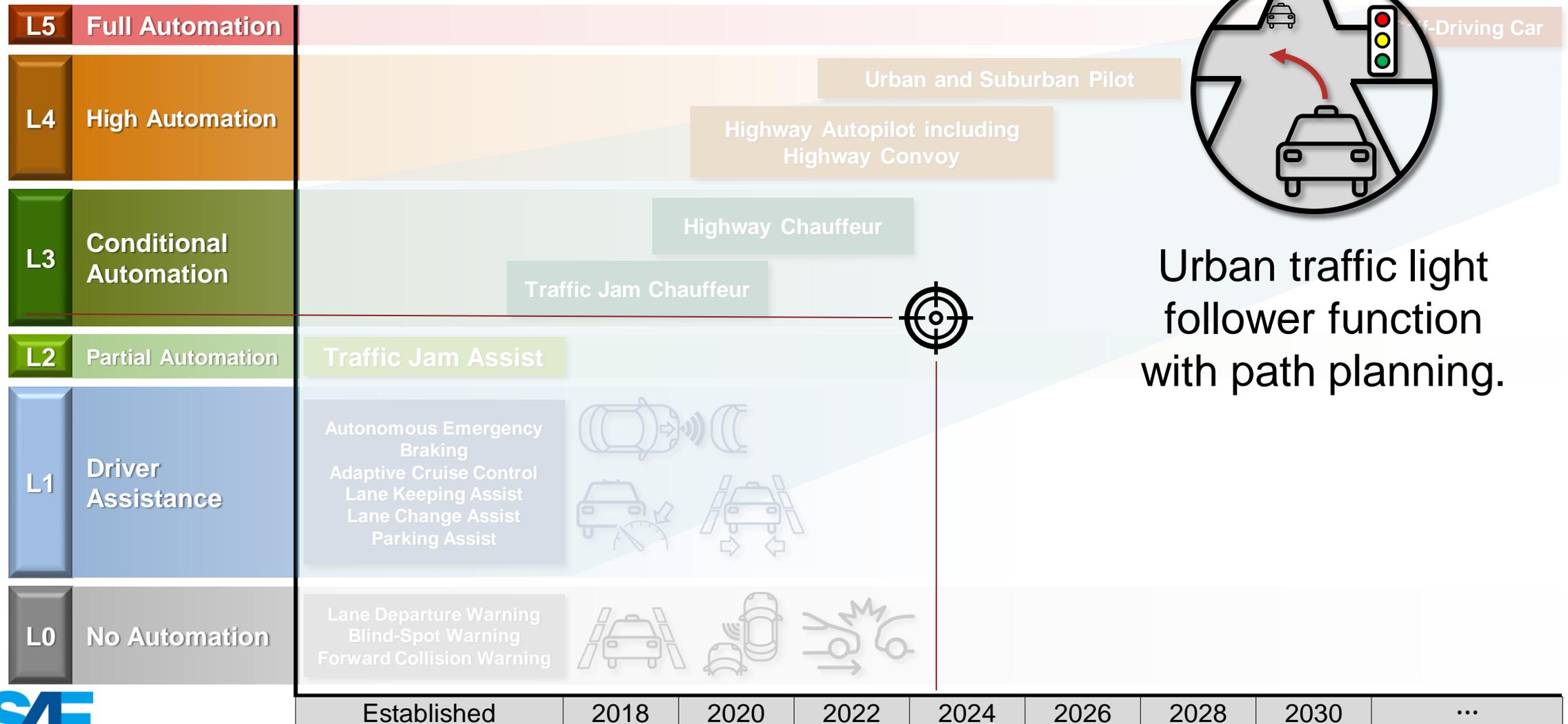
How to...

- ...set up a virtual environment (scene and scenario) with RoadRunner suite.
- ...simulate V2X functionality with Simulink & Stateflow.
- ...setup cosimulation for closed-loop testing with Automated Driving Toolbox.



Primary goal: understand the importance of sophisticated simulation frameworks to instill trust for higher levels of automation.

Evolution of automated driving technology and definition per SAE

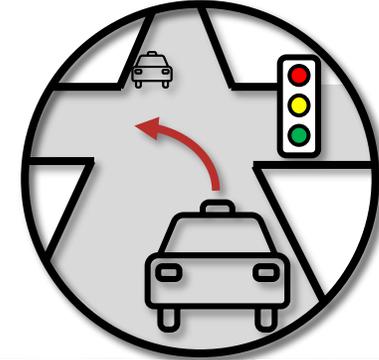


Automated Driving systems tackle a multitude of challenges in an urban environment to ensure safety.

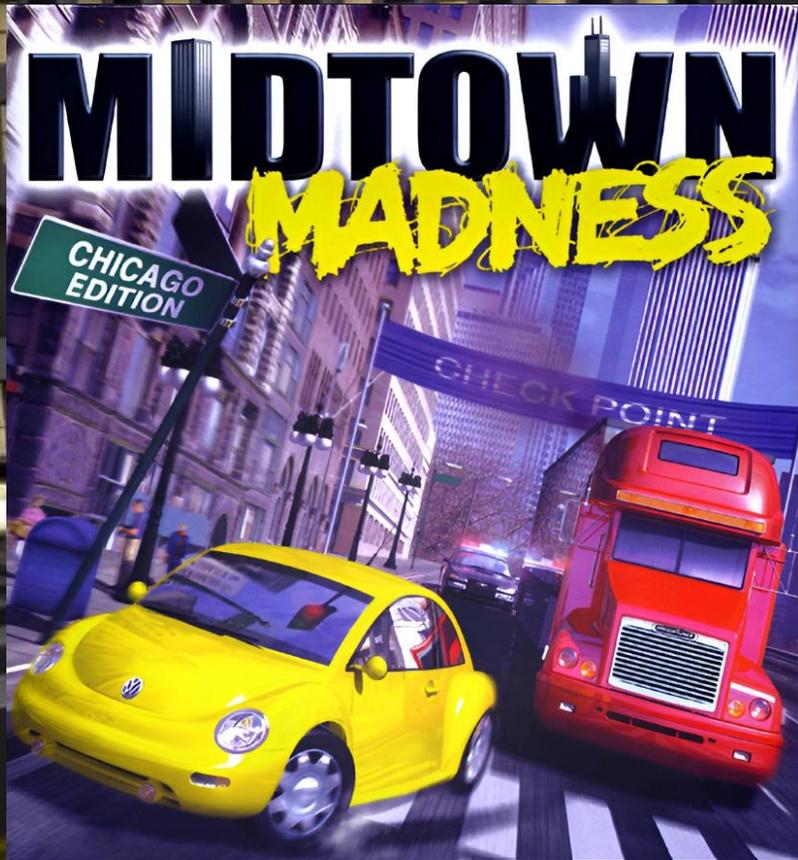
- Complex road geometry, maneuvering in intersections with traffic signals, construction sites, etc.
- Sensor and perception limitations
 - Limited sensor FoV, sensor occlusion
- Communication and connectivity issues
 - GPS signal loss, poor reception

V2X

- Infrastructure enhancement, regulatory compliance, and standard implementation
- V2X standards for e.g., SAE J2735



Traffic light follower at urban intersections



Can someone tell me what this is?

What workflow are we going to learn today?

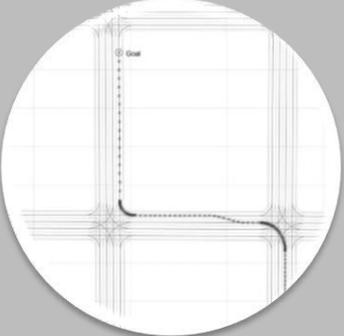


Note) V2X: Vehicle-To-Everything, SPaT: Signal Phase and Timing

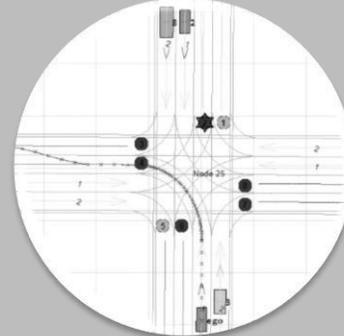
What workflow are we going to learn today?



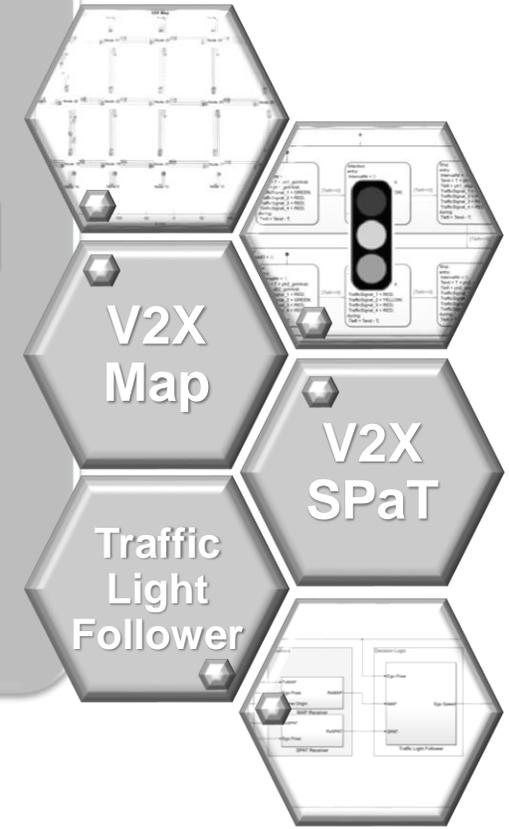
Create
Urban
Scene



Path
Planner



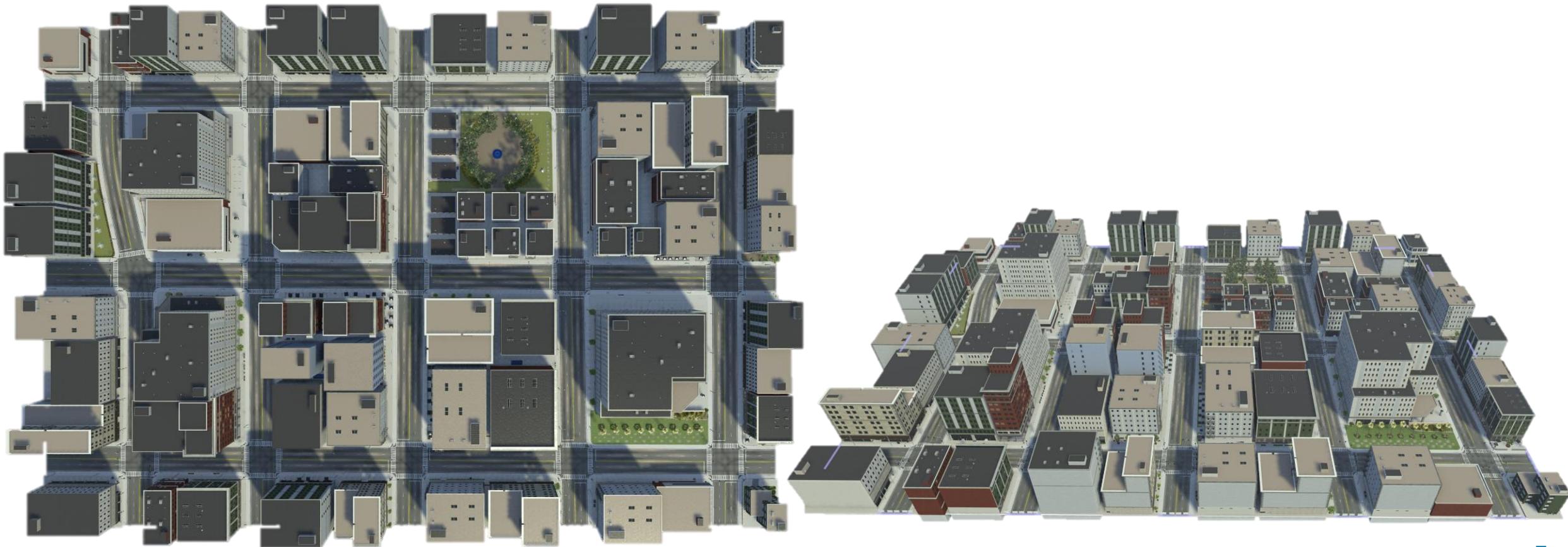
Behavioral
Planner





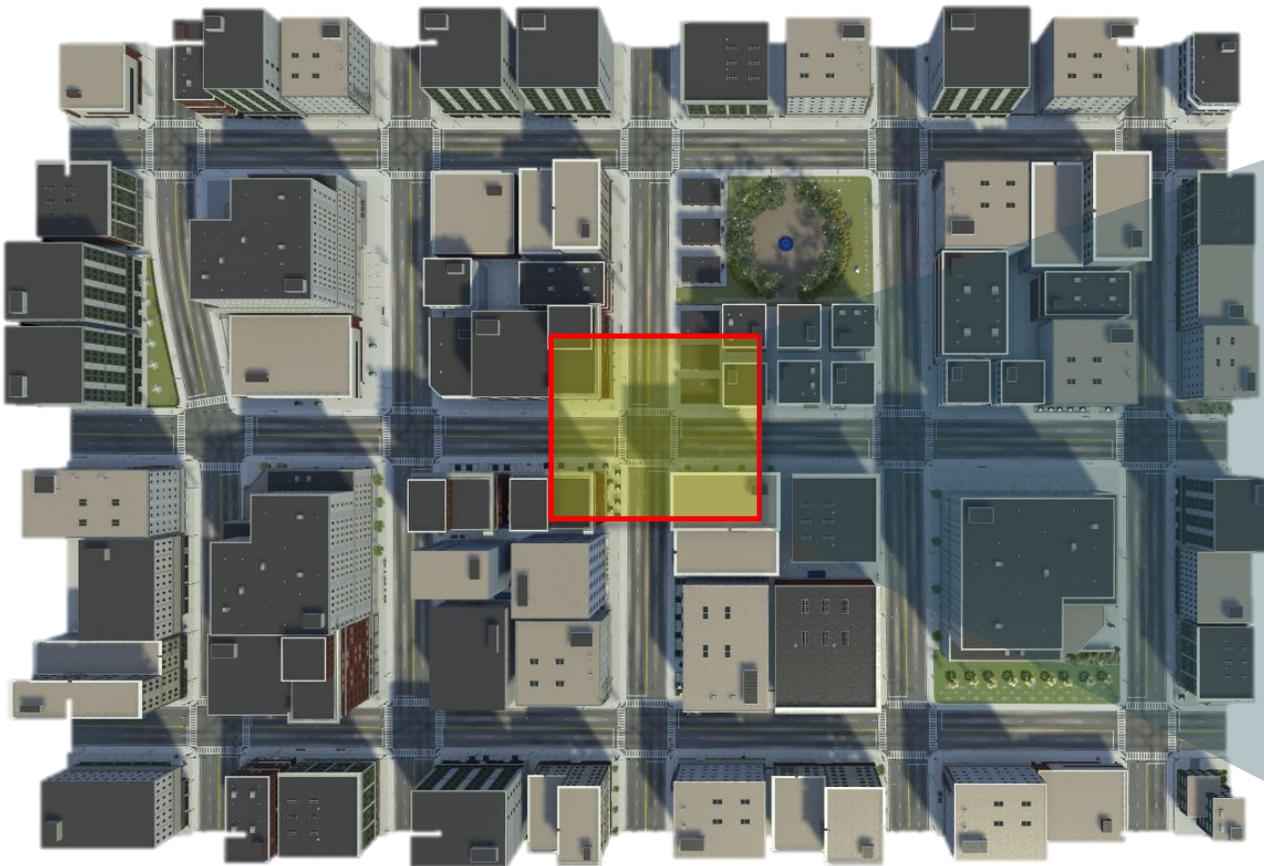
Create a detailed complex urban scene in RoadRunner

- 3D environment of a [US City Block](#) containing 15 intersections with traffic lights.
- All roads in the scene are two-way roads with four lanes.

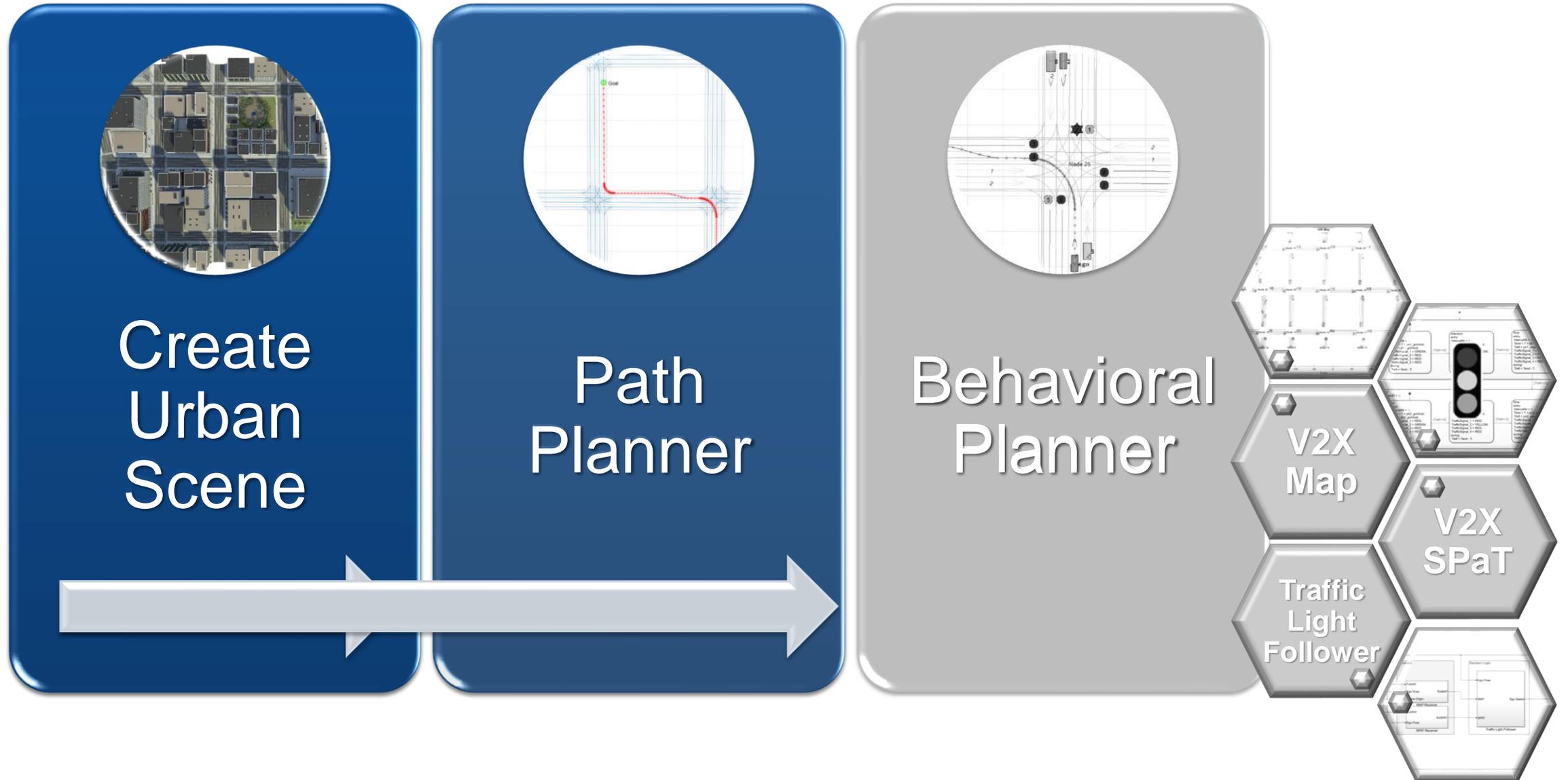


Create a detailed complex urban scene in RoadRunner

- 3D environment of a [US City Block](#) containing 15 intersections with traffic lights.
- All roads in the scene are two-way roads with four lanes.



What workflow are we going to learn today?



Path Planner

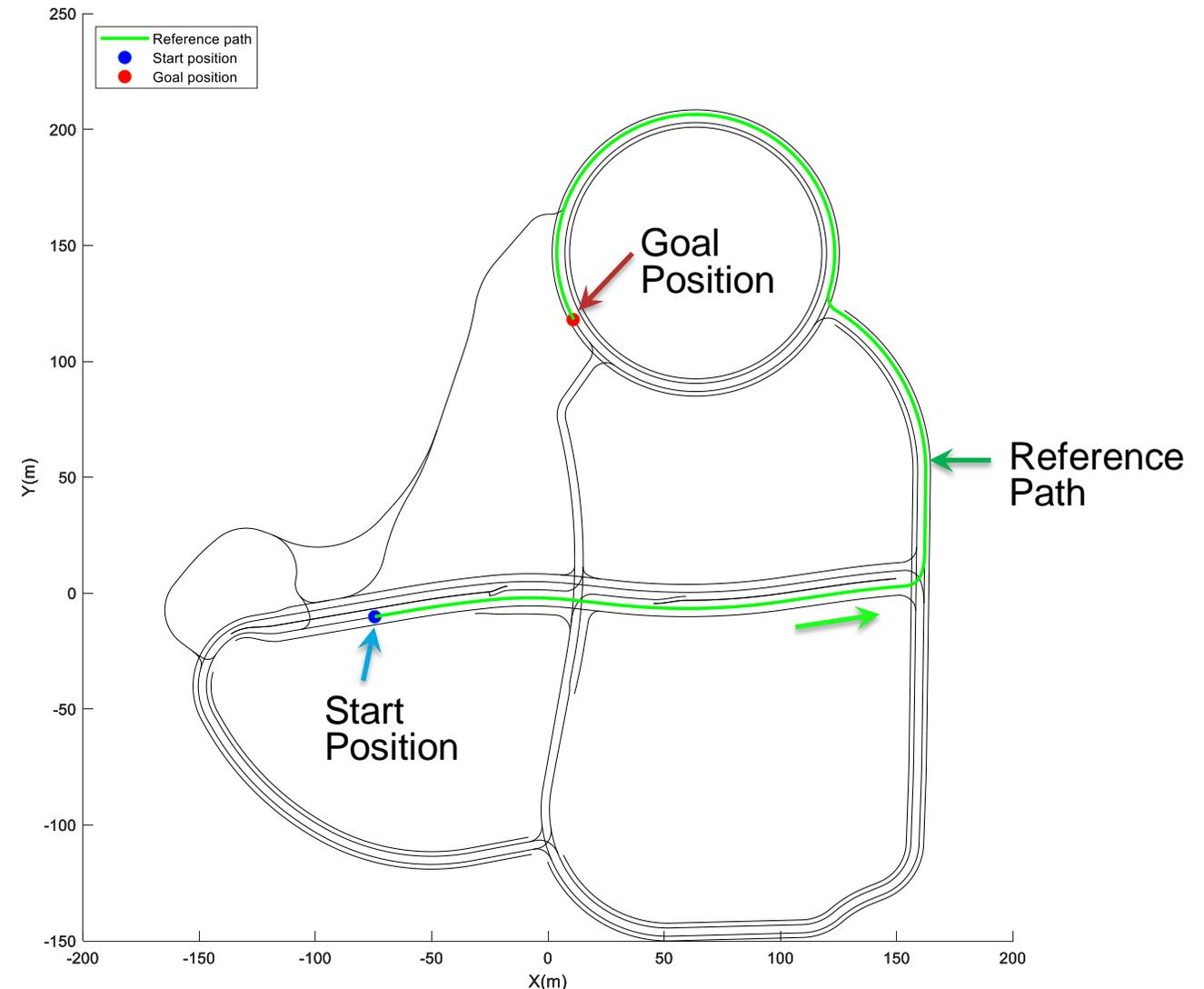


Lane-Level Path Planning with RoadRunner Scenario

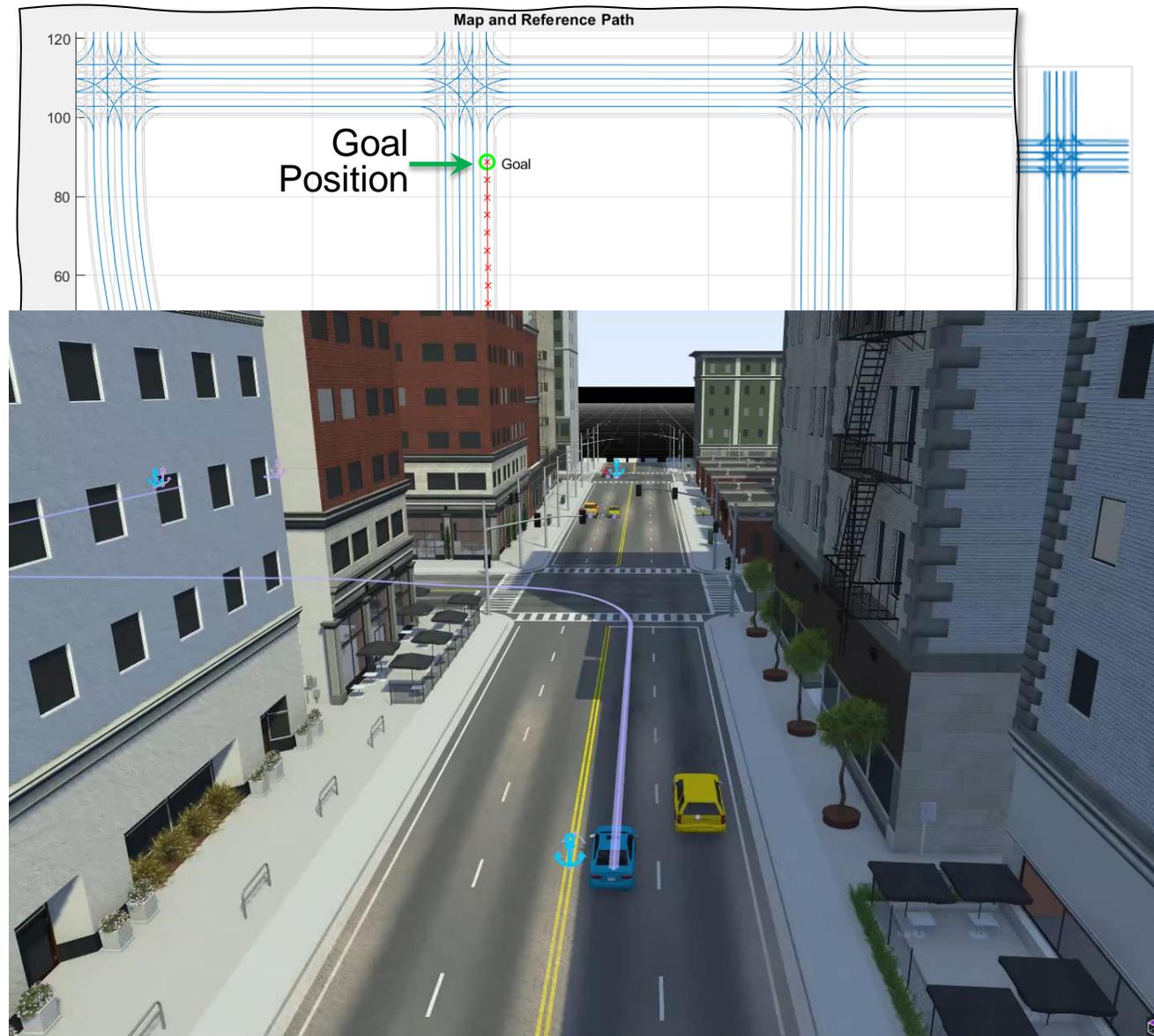
Design a lane-level path planner in MATLAB[®] and cosimulate with RoadRunner Scenario.

- *Automated Driving Toolbox*
- *RoadRunner Scenario*
- *Navigation Toolbox*

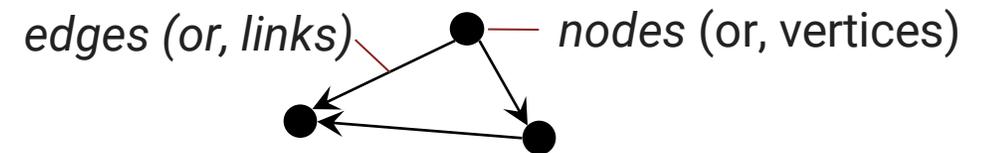
- Finds the shortest path between the start position and the goal position.



Path Planner



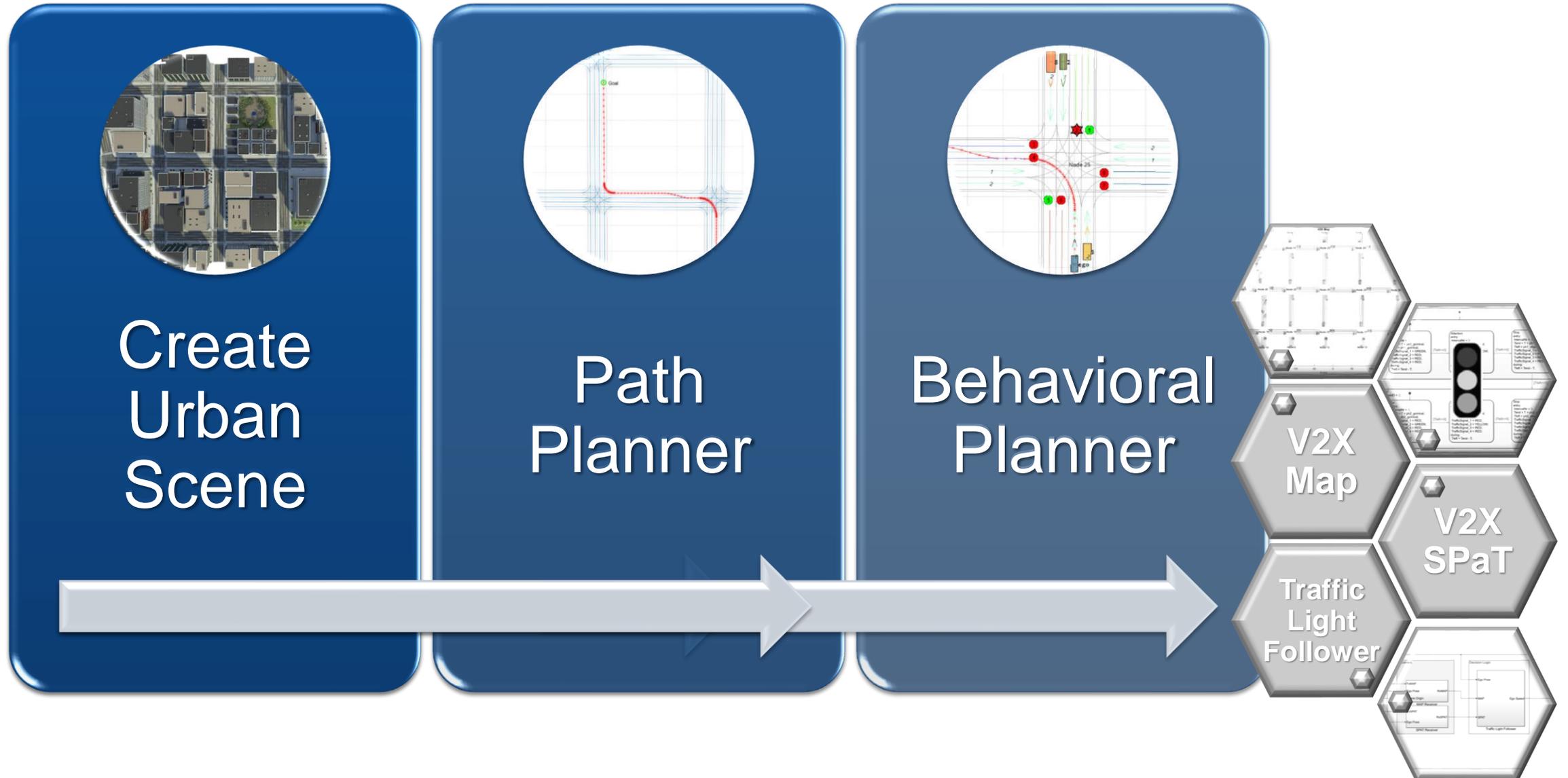
- Nodes and edges table from RoadRunnerHD map



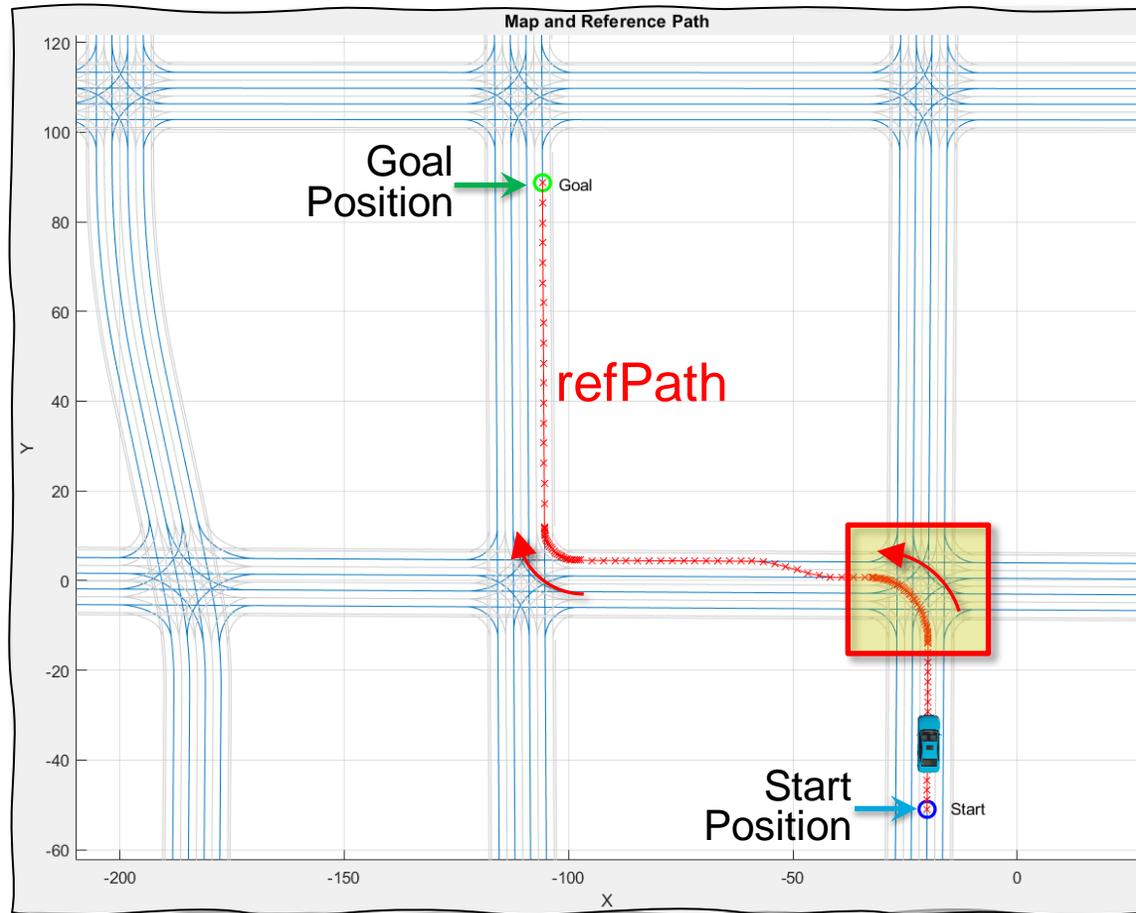
- Graph data structure using [navGraph](#)
- [A* planner](#) from navGraph object
- Shortest path start → goal position.

```
refPath = planPath(planner, graph, Start, Goal);
```

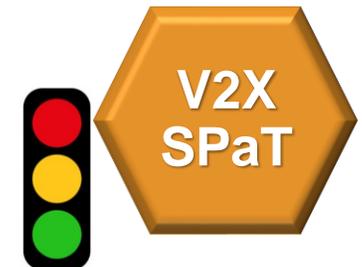
What workflow are we going to learn today?



Behavioral Planer at intersections



- Detect intersection
- State and timing of traffic light
- Appropriate maneuver
- Action decision

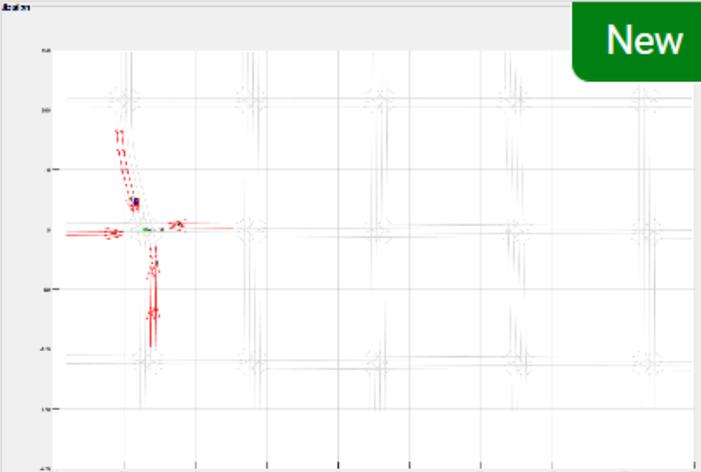


What workflow are we going to learn today?



Generate V2X MAP from RoadRunner

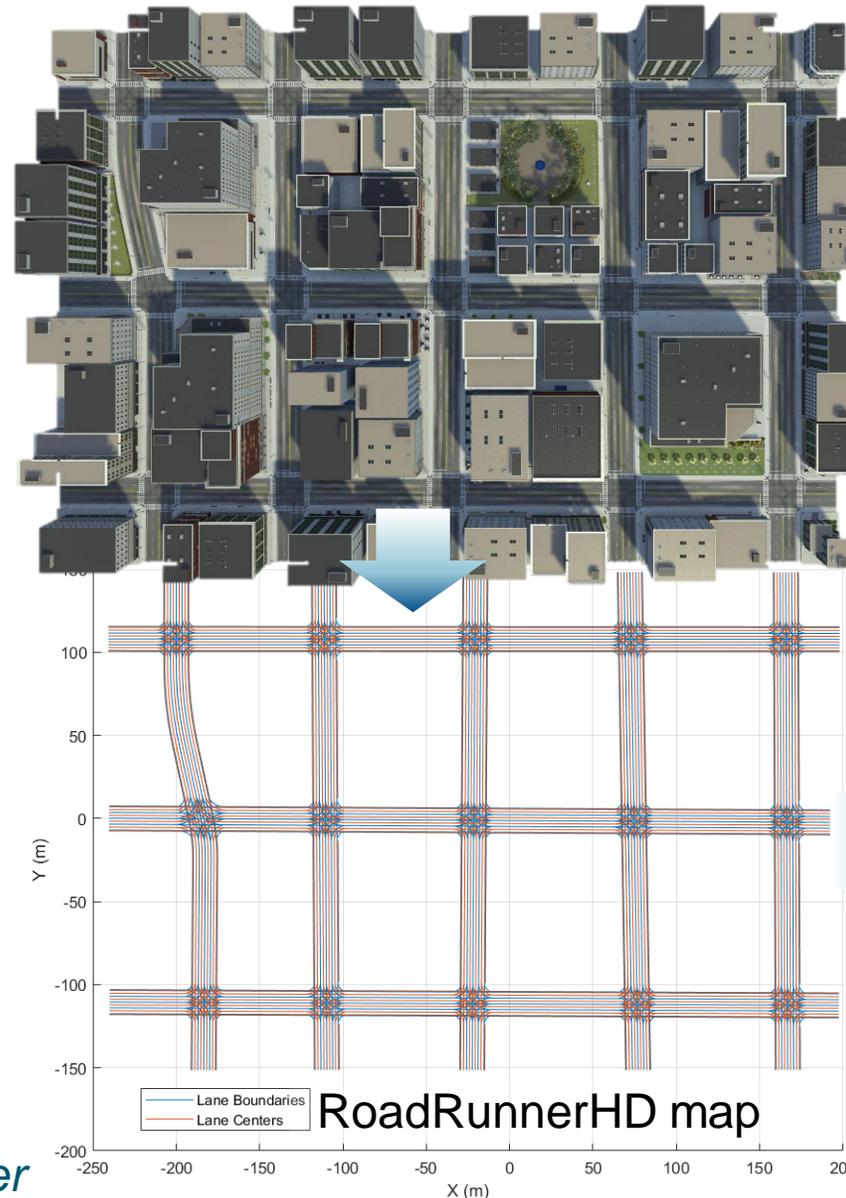
New



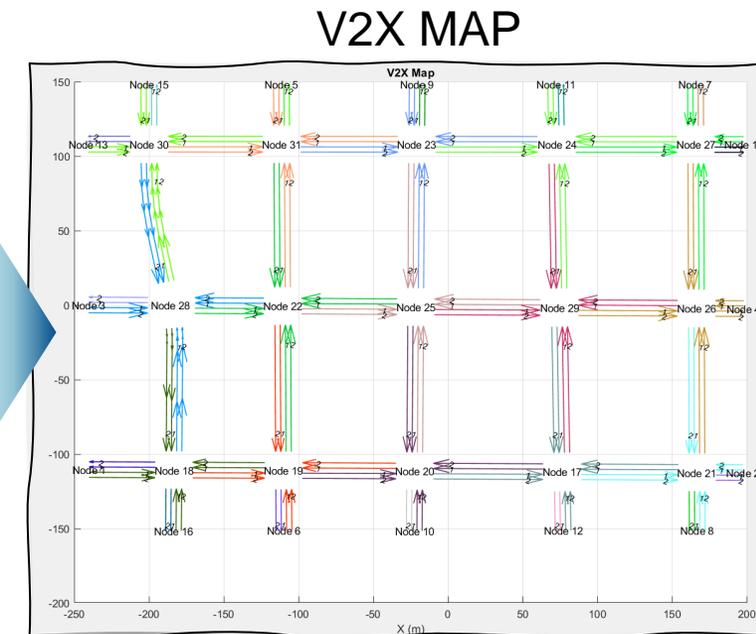
Generate V2X MAP Message from RoadRunner

Generate MAP message and model road side unit for vehicle-to-everything (V2X) communication.

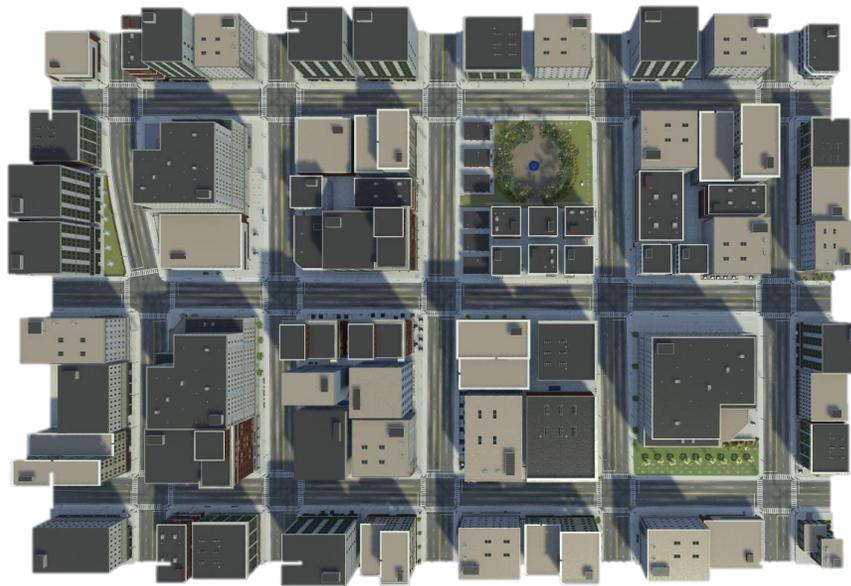
Since R2024a



RoadRunner
Scene & Scenario



Get RoadRunnerHD map for the scene used in scenario simulation



get("Map")



Road data model for representing high-definition (HD) map data in a RoadRunner scene.

Property ^	Value
1x1 roadrunnerHDMap	
Author	""
GeoReference	[0,0]
GeographicBoundary	[-240.8043,-152.9707,-6.8775;198.9754,148.3578,54.5400]
Lanes	2350x1 Lane
LaneBoundaries	2947x1 LaneBoundary
LaneGroups	538x1 LaneGroup
LaneMarkings	3x1 LaneMarking
Junctions	15x1 Junction
BarrierTypes	5x1 BarrierType
Barriers	56x1 Barrier
SignTypes	0x1 SignType
Signs	0x1 Sign
StaticObjectTypes	69x1 StaticObjectType
StaticObjects	2888x1 StaticObject

% Open Scene and Scenario

```
rrApp = roadrunner(rrProjectPath);
```

% Open Scene

```
openScene(rrApp, "USCityBlockBidirectional.rrscene");
```

% Open the scenario

```
openScenario(rrApp, "GenerateMapMessage.rrscenario");
```

% Create Simulation object

```
rrSim = createSimulation(rrApp);
```

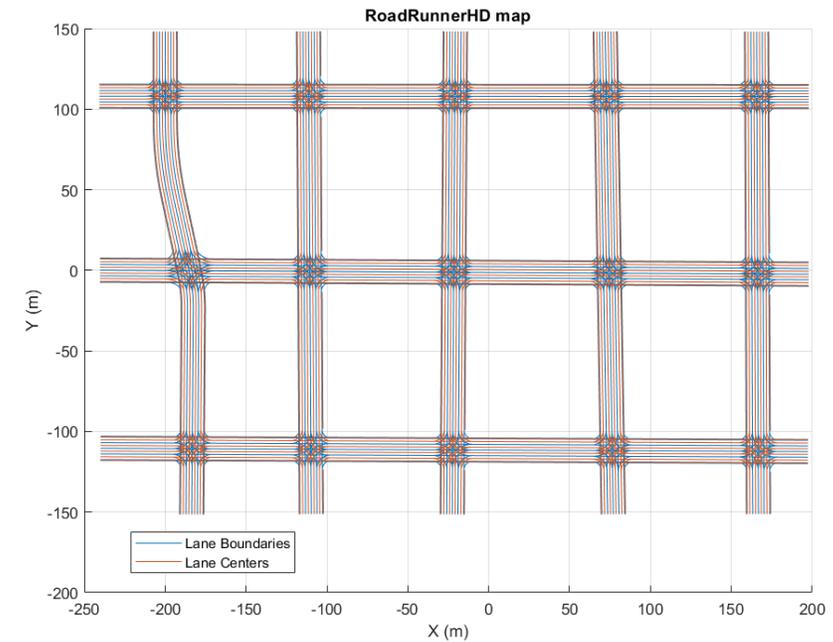
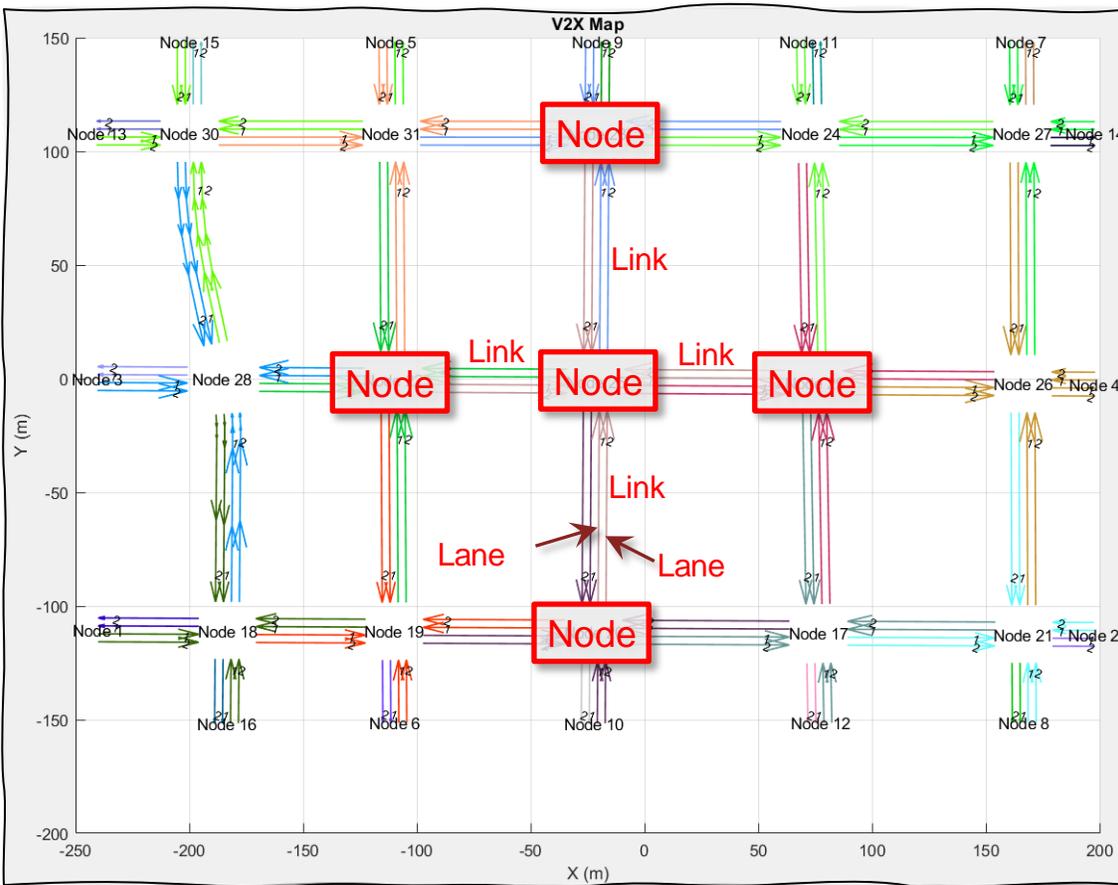
% Get RoadRunnerHD map for the scene used in scenario simulation.

```
rrHDMMap = get(rrSim, "Map");
```

% Plot RoadRunner HD Map

```
plot(rrHDMMap, 'ShowLineMarkers', false);
```

Generate V2X map message from RoadRunnerHD map data

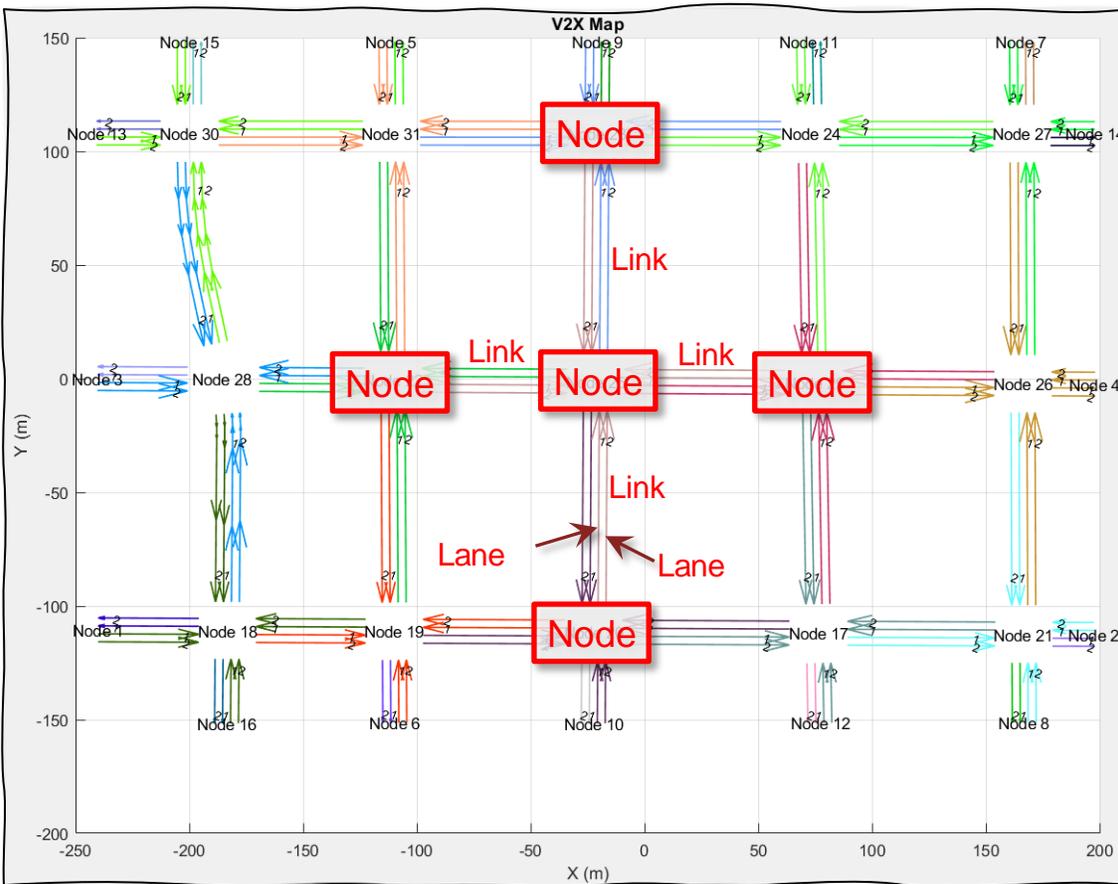


- 1) Finds all **nodes** (intersections or junctions)
- 2) Find **links** connecting all nodes.
- 3) Find connections between all **lanes**.

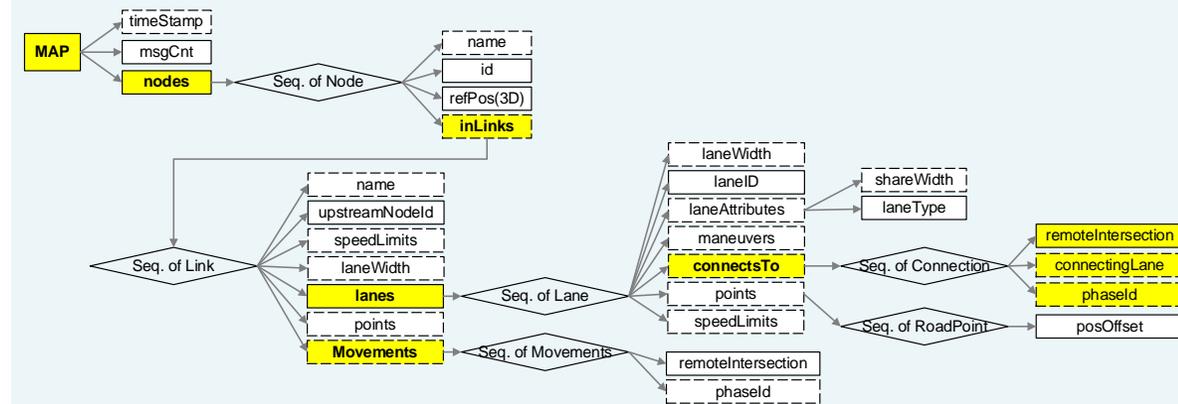
```
% Generate MAP message
sceneOrigin = [42.3648, -71.0214, 10.0];
v2xMapMsg = helperGenerateV2XMap(rrHDMMap, sceneOrigin);

% Visualize MAP message
helperPlotV2XMap(v2xMapMsg);
```

Generate V2X map message from RoadRunnerHD map data



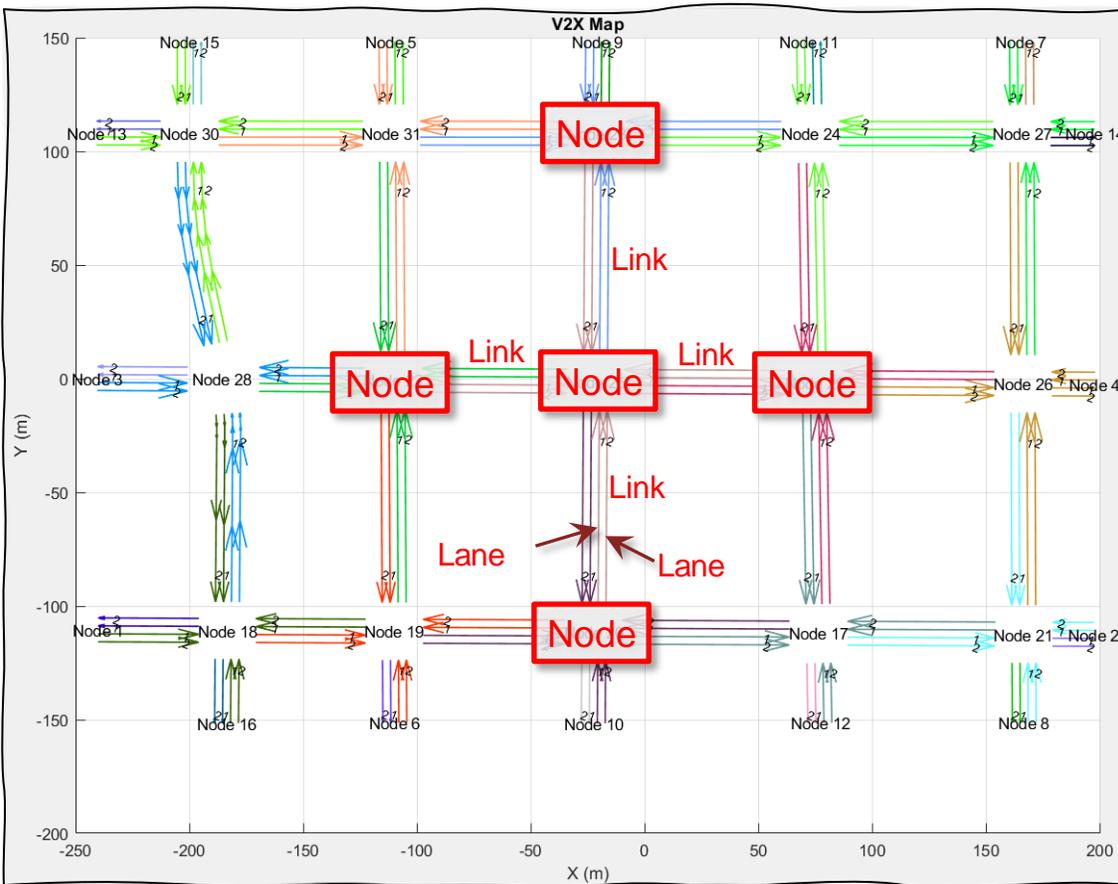
V2X Map message



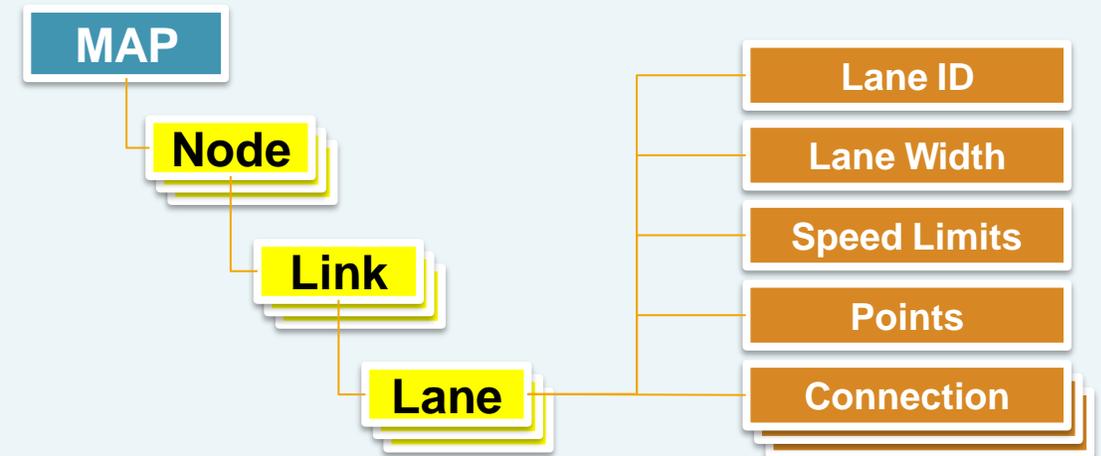
- 1) Finds all **nodes** (intersections or junctions)
- 2) Find **links** connecting all nodes.
- 3) Find connections between all **lanes**.
- 4) Pack the nodes, links, and lane connections with **V2X map messages**.

- **T/CSAE 53-2020**, *Cooperative Intelligent Transportation System — Vehicular Communication Application Layer Specification and Data Exchange Standard (Phase I)*. China Society of Automotive Engineers, 2020.
- **~ SAE J2735**, *V2X Communications Message Set Dictionary*

Generate V2X map message from RoadRunnerHD map data



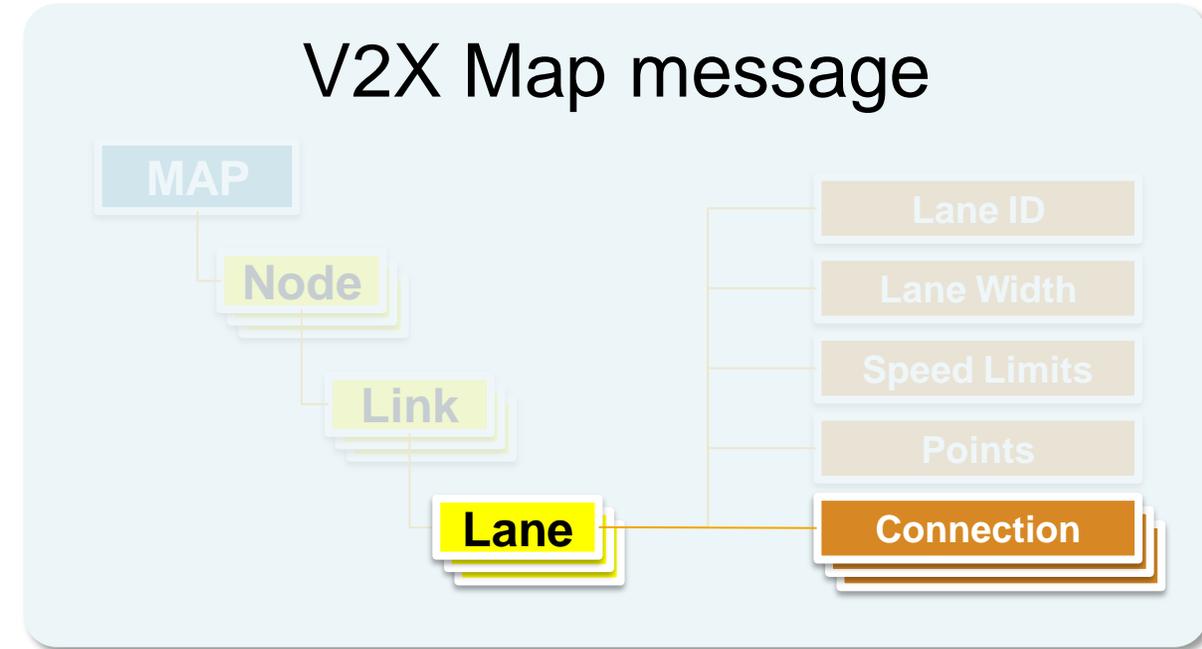
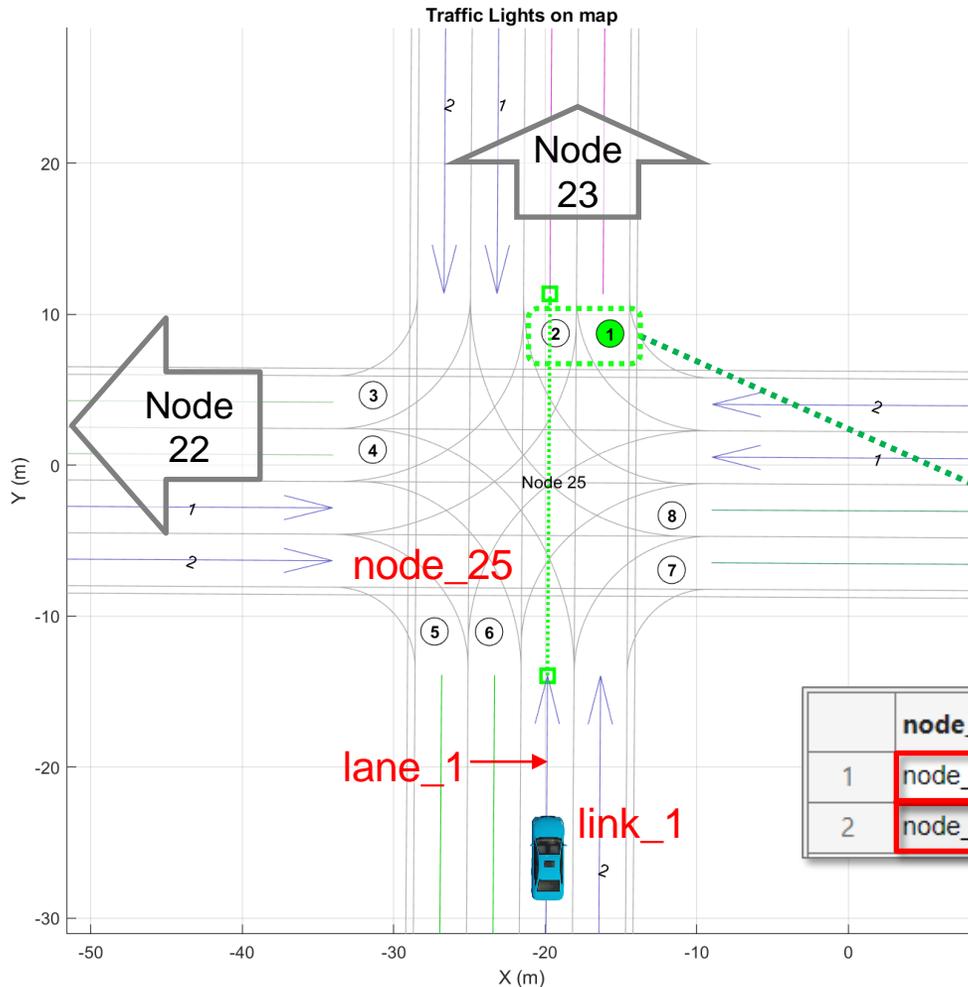
V2X Map message



- 1) Finds all **nodes** (intersections or junctions)
- 2) Find **links** connecting all nodes.
- 3) Find connections between all **lanes**.
- 4) Pack the nodes, links, and lane connections with **V2X map messages**.

- T/CSAE 53-2020, *Cooperative Intelligent Transportation System — Vehicular Communication Application Layer Specification and Data Exchange Standard (Phase I)*. China Society of Automotive Engineers, 2020.
- ~ SAE J2735, V2X Communications Message Set Dictionary

V2X map message: Lane Connection to downstream nodes and traffic signal id



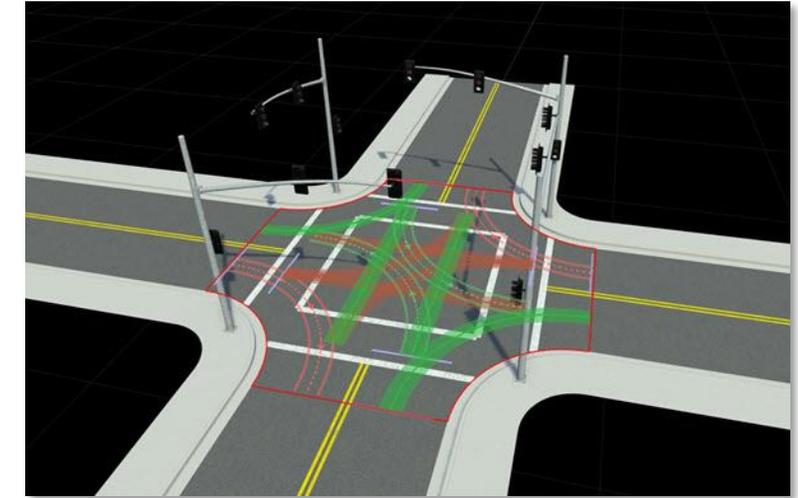
	node_id	link_id	lane_id	signal_id	remote_node	connecting_lane	maneuver	description
1	node_25	link_1	1	2	22	1	0002	Turn Left
2	node_25	link_1	1	1	23	1	0001	Move Straight

What workflow are we going to learn today?

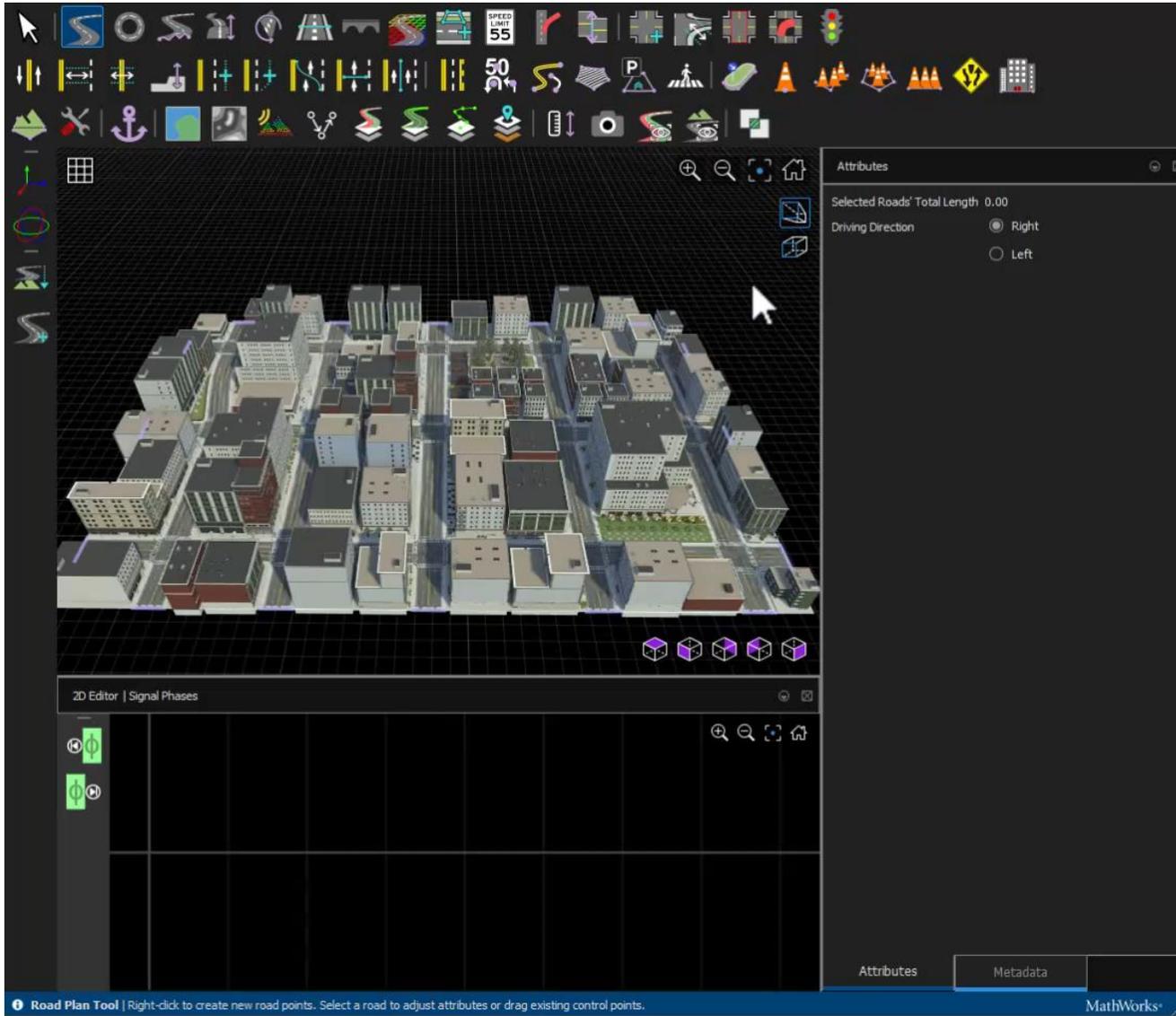
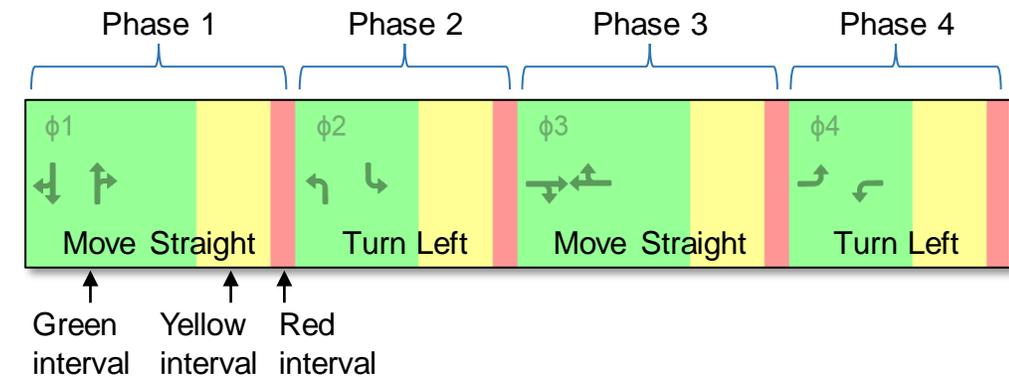


Note) V2X: Vehicle-To-Everything, SPaT: Signal Phase and Timing

“Signal Tool” of RoadRunner



- The Signal Tool is used to configure
 - Junction Signalization
 - Signal Traffic Phases



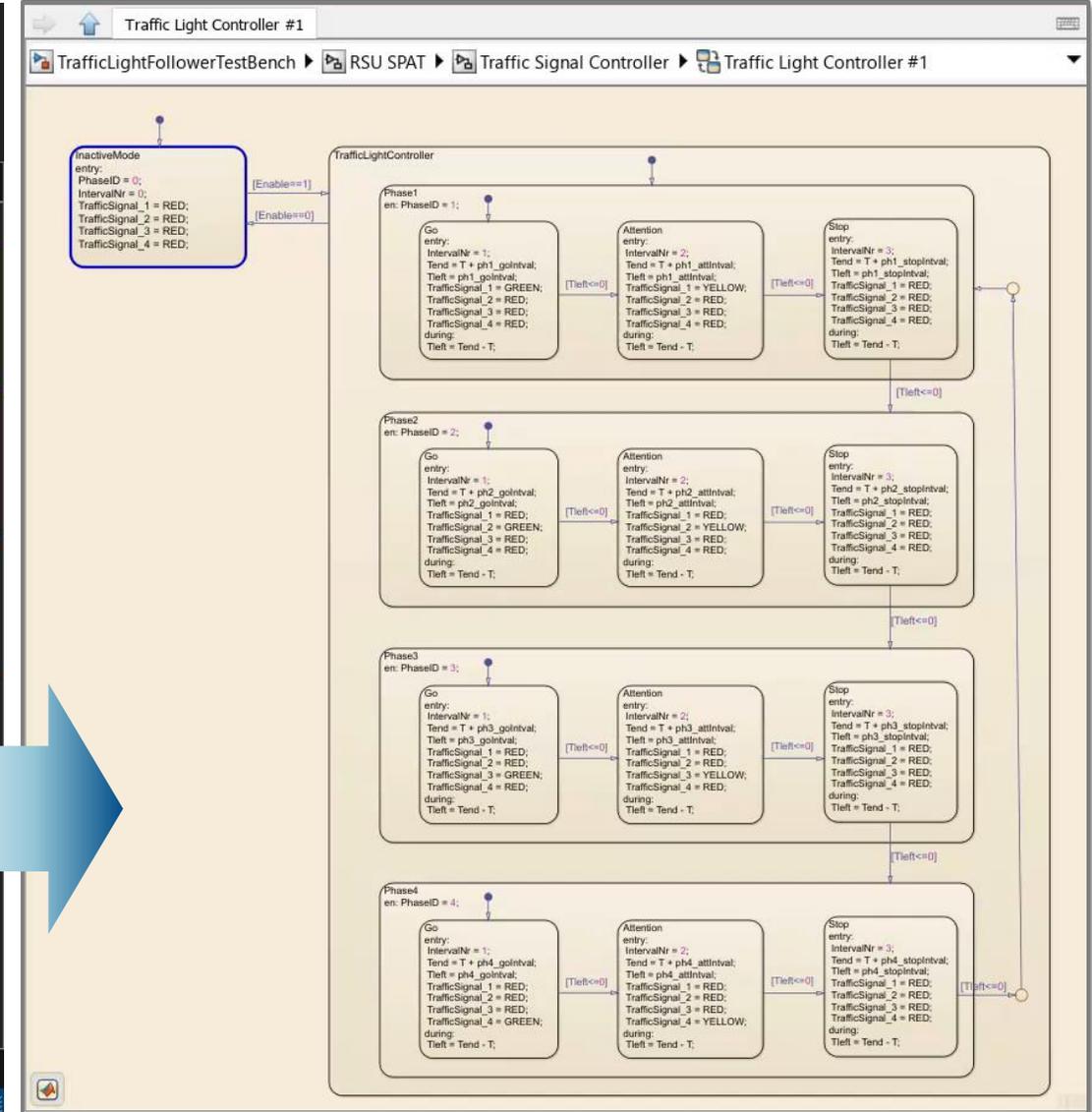
Implement traffic light controller using Stateflow[®]

Attributes

Interval	Interval Name	Interval Type	Interval Time
Interval 1		Green	5.00
Interval 2		Yellow	5.00
Interval 3		Red	1.00

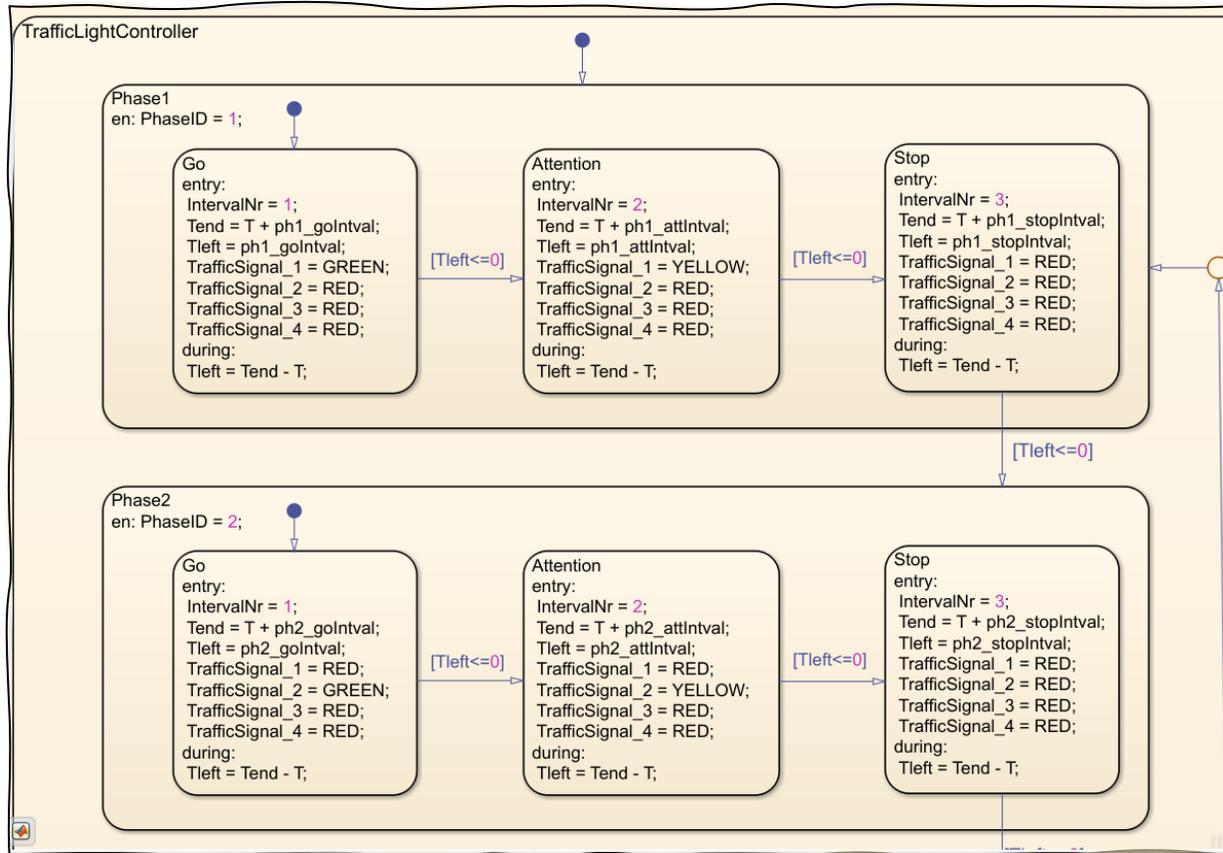
Phase 4

Signal Tool | Select junction to add or edit signal phasing. Select signals to set signal states in current phase.

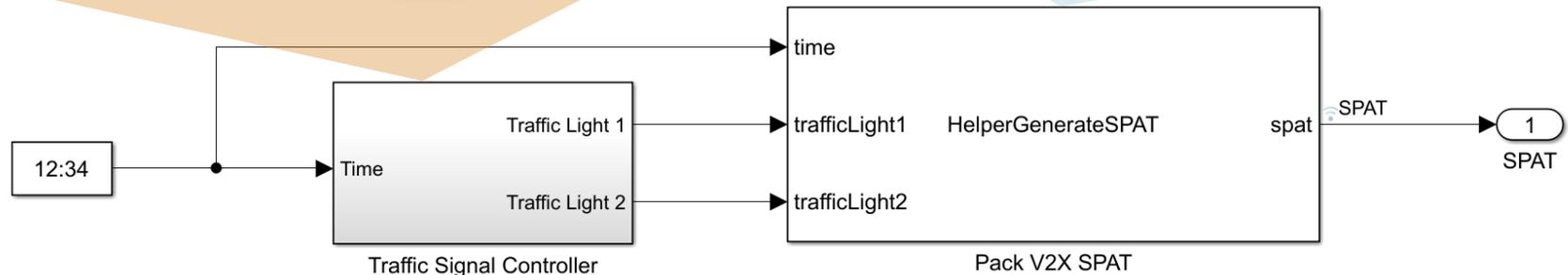
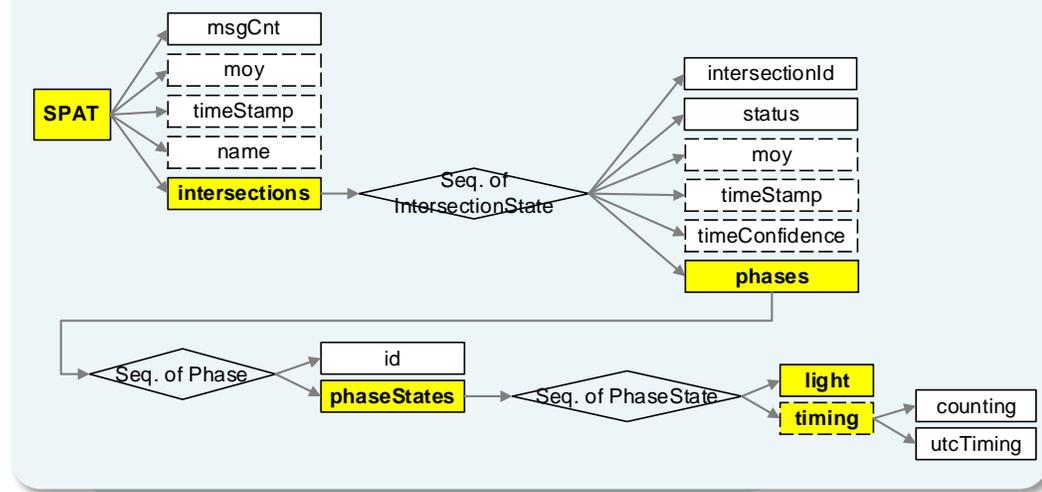


Generate V2X SPaT (Signal Phase and Timing) message

T/CSAE 53-2020

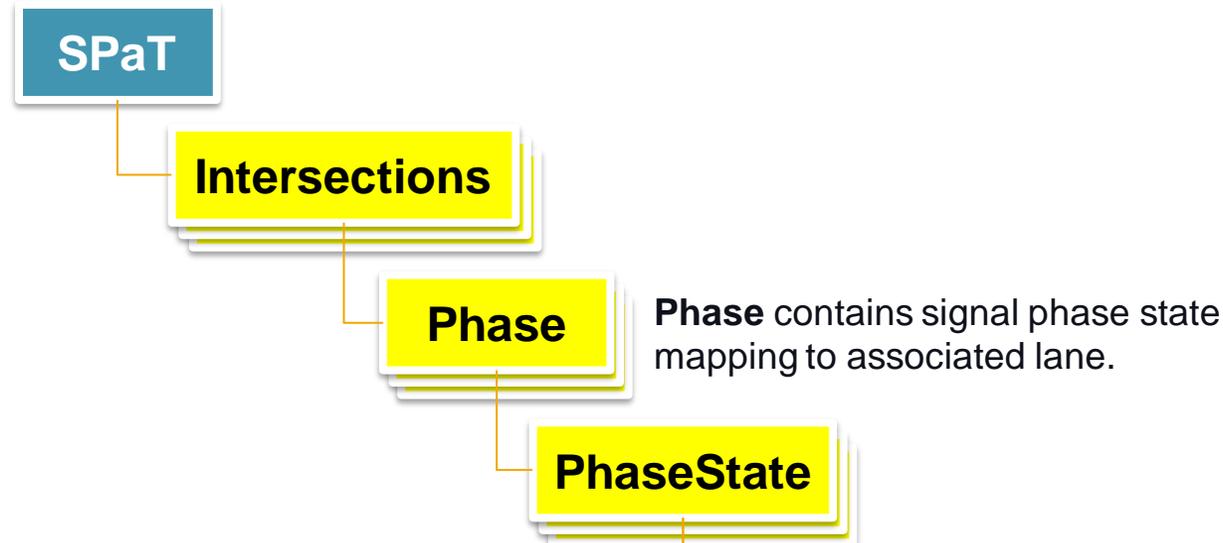


V2X SPaT (Signal Phase and Timing) message



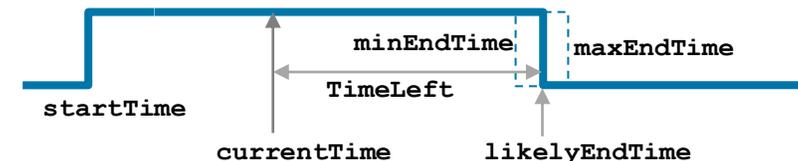
Generate V2X SPaT (Signal Phase and Timing) message

T/CSAE 53-2020

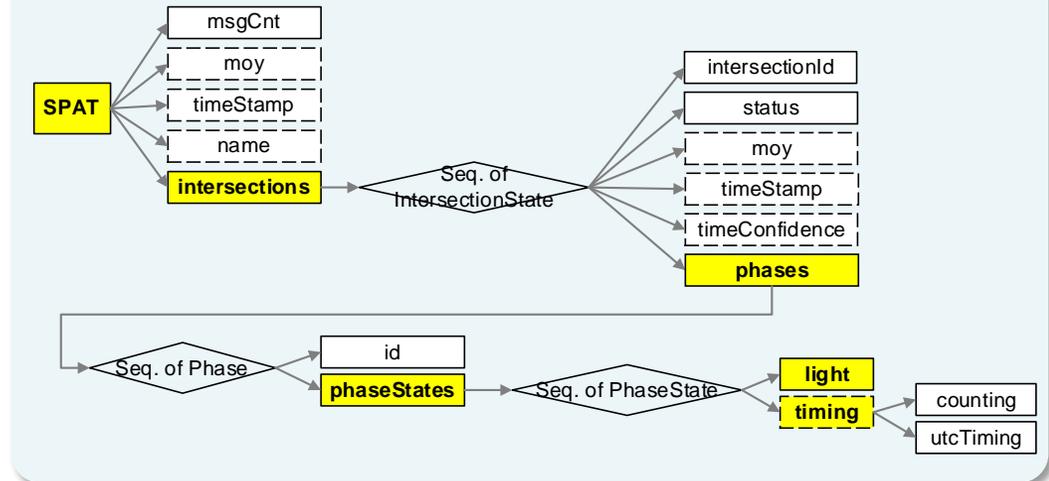


Phase contains signal phase state mapping to associated lane.

Phase state provides a signal phase ID map to an associated lane and **current signal light state and timing** information.



V2X SPaT (Signal Phase and Timing) message



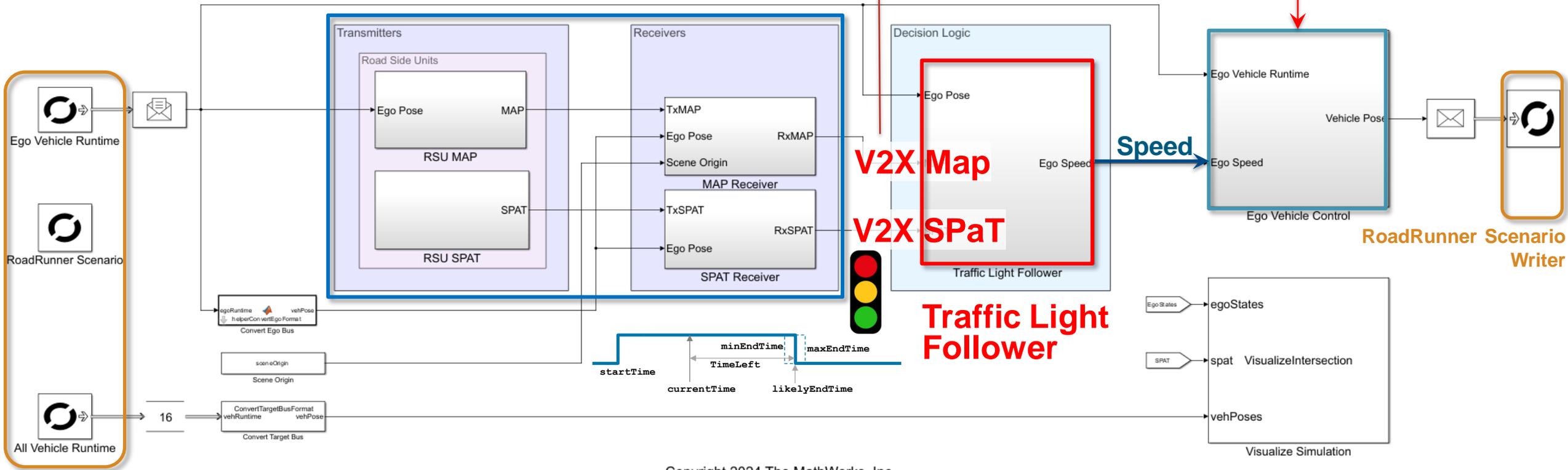
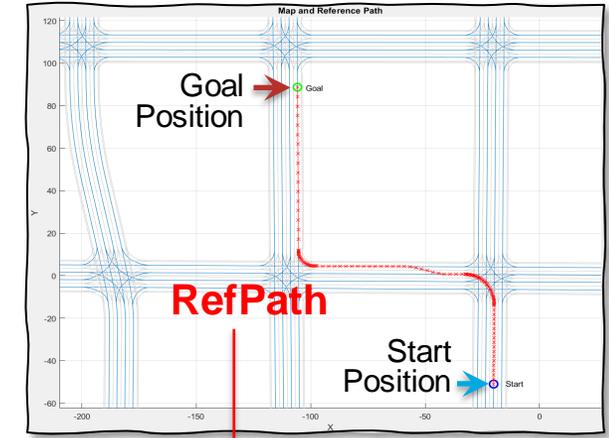
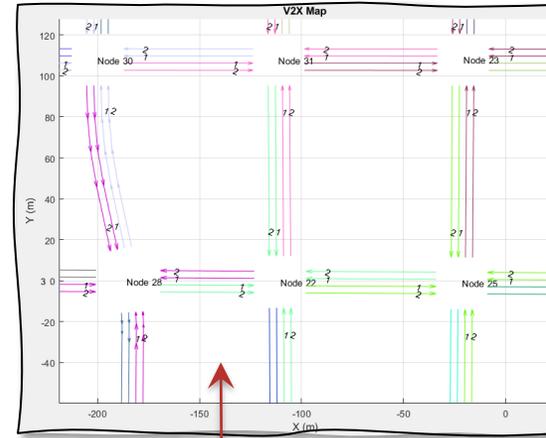
What workflow are we going to learn today?



Note) V2X: Vehicle-To-Everything, SPaT: Signal Phase and Timing

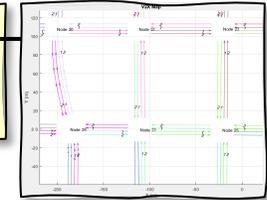
Traffic Light Follower

Road Side Unit for V2X map & SPaT



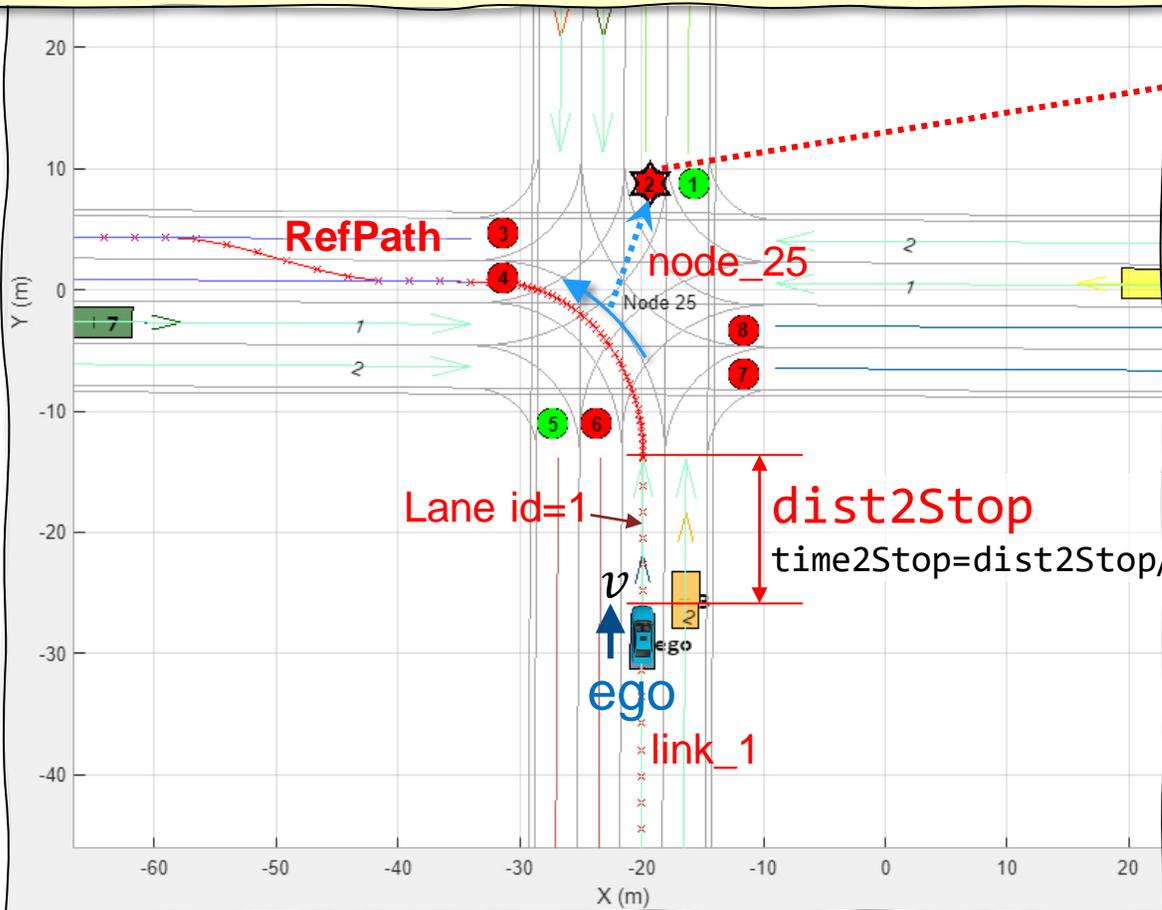
Query V2X Map & SPaT

```
% v2xMapQuery: get map query API object
obj.MapQuery = v2xMapQuery(obj.V2XMap,obj.origin);
```



% **getConnection**: get matching traffic light signal_id and maneuver
% at intersection based on ego current states and reference path

```
[signal_id_ego, maneuver_ego] =
obj.MapQuery.getConnection(lane_ego,obj.RefPath,obj.SearchDist);
```



Ego current state

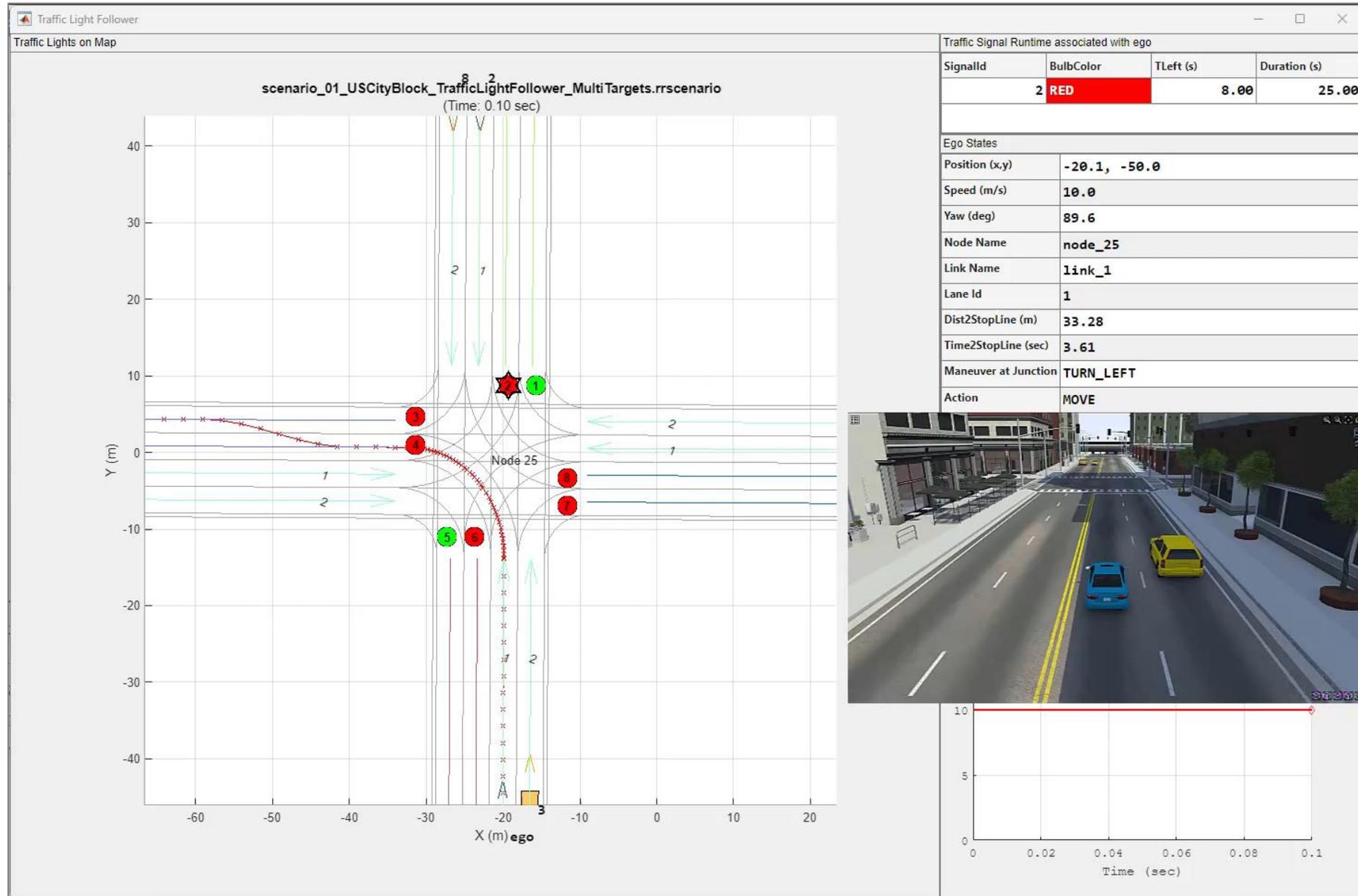


Ego V2X Map info

Maneuver at Junction	TURN_LEFT
Traffic Signal Runtime associated with ego	
SignalId	2
Ego States	
Position (x,y)	-19.9, -29.0
Speed (m/s)	10.0
Yaw (deg)	89.6
Node Name	
Node Name	node_25
Link Name	
Link Name	link_1
Lane Id	
Lane Id	1
Dist2StopLine (m)	
Dist2StopLine (m)	12.28
Time2StopLine (sec)	
Time2StopLine (sec)	1.51
Action	Apply brake to stop

```
% locateVehicleOnMap: get v2x map information
% associated with the current ego position
[node_ego,link_ego,lane_ego,dist2stop,info] =
obj.MapQuery.locateVehicleOnMap(Ego.Position);
```


Traffic light follower with co-simulating RoadRunner Scenario



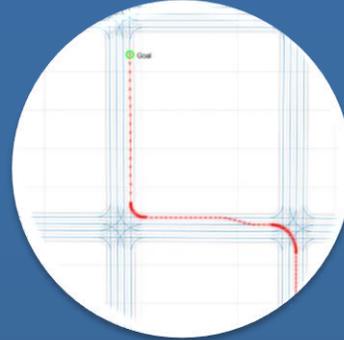
Key Takeaways:

Automated Driving in the Urban Environment with RoadRunner Scenario



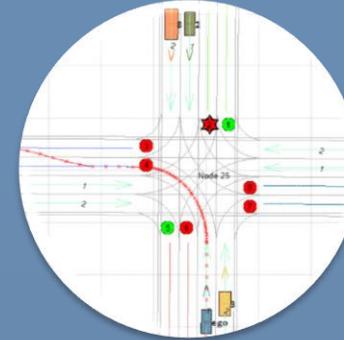
Create a complex **Urban Scene** consisting of Intersections with Traffic Lights.

- *RoadRunner*
- *RoadRunner Asset Library*



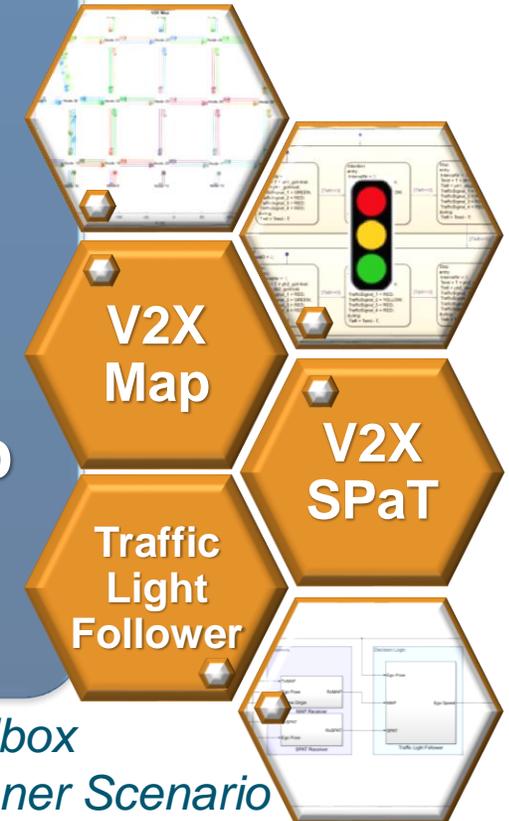
Design **Path Planner** using **A-star Planner**.

- *Navigation Toolbox*



Develop **Behavioral Planner** using **V2X Map and SPaT**.

- *Automated Driving Toolbox*
- *RoadRunner, RoadRunner Scenario*
- *Simulink, Stateflow*



MathWorks
**AUTOMOTIVE
CONFERENCE 2024**
Europe

Thank you

Please contact me at avalluri@mathworks.com
with questions

