# Basic use case: networked traffic control for autonomous cars.

The project is working on how to deal with traffic management in an intersection.

The emergency cars (like ambulance, police car), pedestrian, and ordinary vehicles may pass the intersection.

Assumption:

1. Each vehicle and pedestrians equipped a smart device to send intersection pass request via that, receive permission to pass or not, and define the origin and destination.
2. Traffic control system controls the traffic flow and implement the logic of decision-making.
3. There is a central management unit to set priorities and manage traffic to prevent collision.
4. The priority works like this in general:

Emergency Car>Pedestrian>Ordinary Vehicles

1. For the number of cars less than 4, the road, which has a lower time to achieve, has the higher priority.
2. For cars more than 4 in a queue, we put higher priority.
3. Turning each car to its right direction is always feasible.
4. There are numerous possible concurrency of passing the intersection. However, we focused on a few numbers to implement and show the logic that is going to work in general.

|  |  |
| --- | --- |
|  |  |
| No collision happens – pass simultaneously | No collision happens – pass simultaneously |
|  |  |
| Collision happens – prioritizing needed | Collision happens – prioritizing needed |

Some cases that are implemented are as below:

* 1. Two ordinary cars passing simultaneously
     1. cars go straight on horizontal or lateral direction – without collision
     2. cars are turning to their left direction on horizontal or lateral direction – potential collision case, prioritizing is needed
  2. One emergency and one ordinary car pass request
  3. Pedestrian and ordinary car pass request

A truth table illustrates the logic better. If car1 in on the North, and car2 is on the south, one probable truth table goes as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Car1/Car2 | North-South | North-West | North-East |
| South-North | Simultaneously without collision | Collision happens  prioritizing needed | Simultaneously without collision |
| South-West | Simultaneously without collision | Collision happens  prioritizing needed | Simultaneously without collision |
| South-East | Collision happens  prioritizing needed | Simultaneously without collision | Collision happens  prioritizing needed |

# Specification of the analysis model in SysML

## Specify at least 10 requirements

1. **Braking Distance *β***
   1. The Braking Distance ***β*** is calculated based on the Current Velocity ***V*** and Vehicle Type ***T.***
2. **Brake Pressure *μ***
   1. The Brake Pressure ***μ*** is calculated based on the Braking Distance ***β*** andVehicle Type ***T.***
3. **Automatically Driven Vehicles**
   1. The Vehicles Send and Receive the Braking Distance ***β*** Signal with respect to Time.
   2. The Brake Pressure ***μ*** is estimated by Current Velocity ***V*** and Braking Distance.
4. **V2I Communication**
   1. The vehicles in the close vicinity communicate their Current statistics (Geo-coordinates, Velocity/Speed, Route Map, Headed To, and Hault Status) and Route map to a Localized Server.
   2. Vehicles may convey an SOS signal and Pass-by Signal to Ego Vehicles (to demand way/overtake).
5. **Vehicle Type *T***
   1. Vehicle type **T** is defined by the physical characteristics of the Vehicle such as Height *H*, Length *L*, Width *W,* and Utility *U* (*SUV, MUV, Sedan*) or Emergency Services (Fire Truck, Ambulance).
   2. The Power Train characteristics of the vehicle (Engine Power in *HP*, Cylinder capacity in *CC*, Fuel Type [*Petrol, diesel, Gas*] ) also vary based on the Vehicle Type ***T****.*
6. **Road type *R***
   1. Road Type ***R*** is defined by the Urban Planning Database.
   2. The Road Type signifies the Maximum Permissible Speed Limit ***L*** in *KMPh.*
7. **Distance to Crash *ω***
   1. The Distance to Crash ***ω*** is calculated based on the values of Distance sensor (on both sides Front & Back ).
   2. This is the minimum maintainable distance between two vehicles without causing any accidents.
8. **Collision Avoidance Algorithm (CAA)**
   1. The vehicle will Calculate the Distance to Crash ***ω*** (distance between the immediate Ego vehicle and the immediate Rear Vehicle) with the help of a Distance sensor (on both sides Front & Back ).
   2. The vehicle will maintain a Safe Distance ***h*** between the Ego vehicle and itself to avoid contact/crash.
   3. This distance *h* is subject to vary based on Vehicle Type ***T***(SUV, Sedan, MUV) and road type *R* (Metro/Highway).
9. **Maximum Achievable Speed *A***
   1. Maximum Achievable Speed ***A*** is calculated based on Vehicle Type ***T,*** Distance to the Destination ***D,*** and current traffic conditions.
10. **Time to Destination *δ***
    1. The Time to Destination ‘*δ’* is calculated based on the Distance to Destination ***D*** from the Maps.
    2. Real-Time Traffic Model is used to bias(+/-) the Time to Destination ‘*δ’*.
    3. Vehicle Type T is used to decide the Maximum Achievable Speed ***A*** for certain vehicles and Location Specific Speed Limits (Schools, Hospitals, and Highways).
11. **Maximum Permissible Speed Limit *L***
    1. All vehicles are fitted with Speed Governors and the speed data is pushed to the localized cloud continuously.
    2. The maximum achievable speed S is determined by the Distance between Ego Vehicle *h*, Road Type *R*, and Time Required to reach destination *T.*
    3. Any Vehicle achieving speed beyond the Maximum Permissible Speed Limit ***L*** will be centrally imposed with a fine digitally.
12. **Emergency Flag *E***
    1. Every vehicle has a right to publish an Emergency Flag ***E*** if it has to bypass the standard regulations (Maximum Permissible Speed Limit *L*) in case of an emergency.
13. **Assigned Vehicle On-Road Priority *Ѳ***
    1. Every vehicle has its predefined Vehicle Priority Ѳ set.
    2. Vehicle Priority Ѳ is defined based on Vehicle Type T (Fire Truck, Ambulance) and Emergency Flag ***E.***
14. **Intersection Management System**
    1. The Intersection Management System will use the planned route maps of the vehicles, V2V Communication, Distance sensors, and all the above-mentioned parameters to manage the intersections efficiently.

## Define the context / use cases of the system

### Refine three use cases with activity diagrams

## Define the analysis architecture (Context Diagram) with block diagrams (at least 10 blocks) and describe the main interaction with the environment with sequence diagrams.

# Specification of the analysis model in SysML

## Constraint the system with parametric constraint diagrams (at least 5 constraints)

* Number of cars approaching to the intersection in one direction (less than 4 or more than 4) more than four cars in one direction; switch the case to traffic jam. Because we assume that each direction on each side of the intersection has three lines. So more than it, is considered as congestion status.
* Safe distance from intersection in case to suggest the replacement (less than 1km and more than 1km) send the speed down command to the cars after 1km in case of presence of any pedestrian passing the intersection.
* Time to achieve the intersection calculated from speed, which is used for prioritizing. Timestamps works like which car passes by sensors first it has priority in case of same number of cars in 2 sides.
* Cars directions calculated from origin and desired destination (Boolean value)
* Emergency flag(Boolean value)
* Pedestrian flag (Boolean value) - If emergency flag and pedestrian flag is zero so this is an ordinary situation
* Cars turning status, left, right, straight (Boolean value)

## Allocate the diagrams to each other in an allocation diagram

# Based on the analysis model derive a design model

## Refine the architecture (block diagrams) with the help of internal block diagrams (add at least 5 blocks)

## Define the behavior with state machines of at least 3 components

### Show / argument that the state machine behavior fulfills (refines) the interaction behavior (as modeled in the analysis phase)