A1 Conceptual Architure of Apollo

Cisc 322 Group 20

YouTube: https://youtu.be/Ud1Wr7h7Yss

Intro

Members:

Yucan Li (Planning, Sequential Diagram, Group Leader) 18yl259@queensu.ca

Yuzhe He (Perception, Prediction) 18yh46@queensu.ca

Xuchuan Mu (Monitor, HMI) 18xm24@queensu.ca

Yiming Zheng (Control, CANBus, PowerPoint Editor) 19yz38@queensu.ca

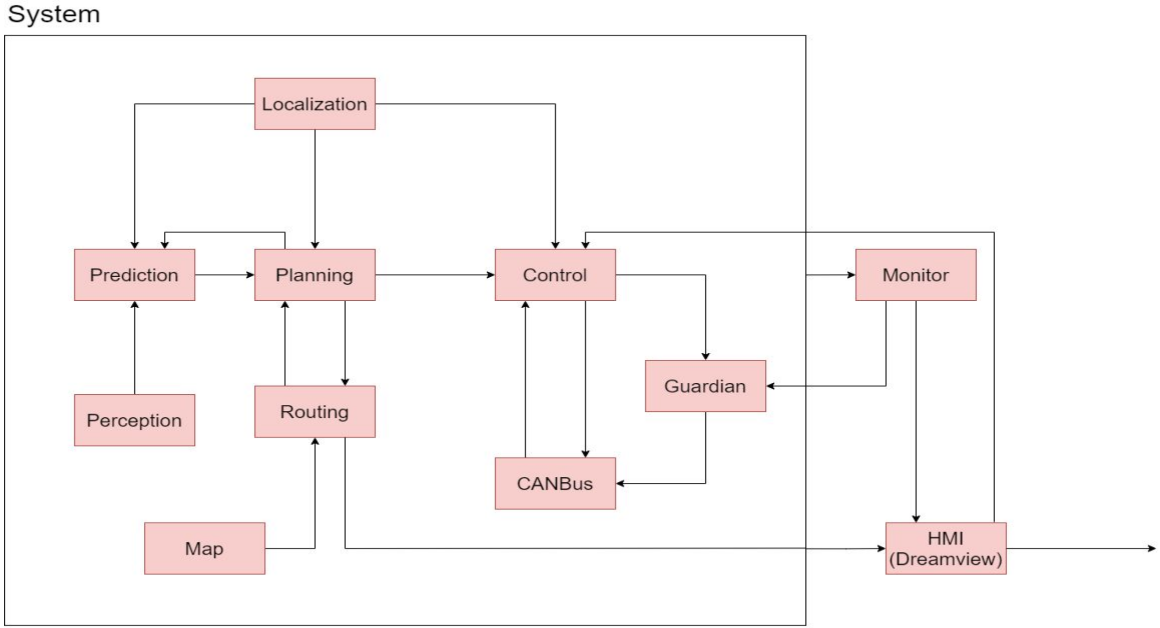
Wenran Hou (Localization, Video Editor) 18wh10@queensu.ca

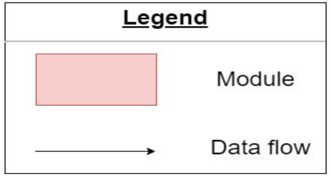
Mukun Liu (Map, Routing) 19ml13@queensu.ca

Apollo

Apollo is an open-source platform which enables autonomous driving developed by Baidu. We are focusing on several functional modules inside Open Software Platform including Map Engine, Routing, Localization, Perception, Prediction, Planning, Control, CANBus, Guardian, Monitor and HMI.

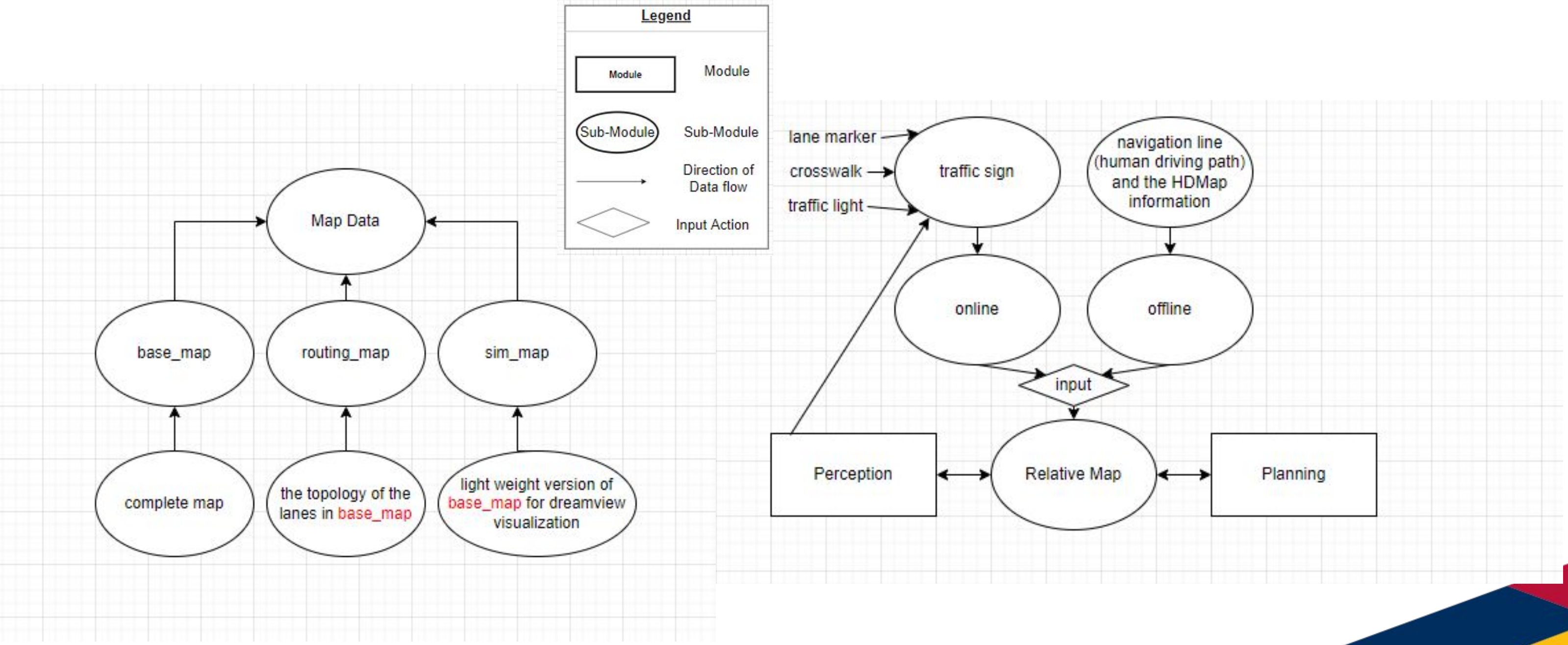
Conceptual Architecture





Map

Goal: load the map and provide a series of API for others to use.



Routing

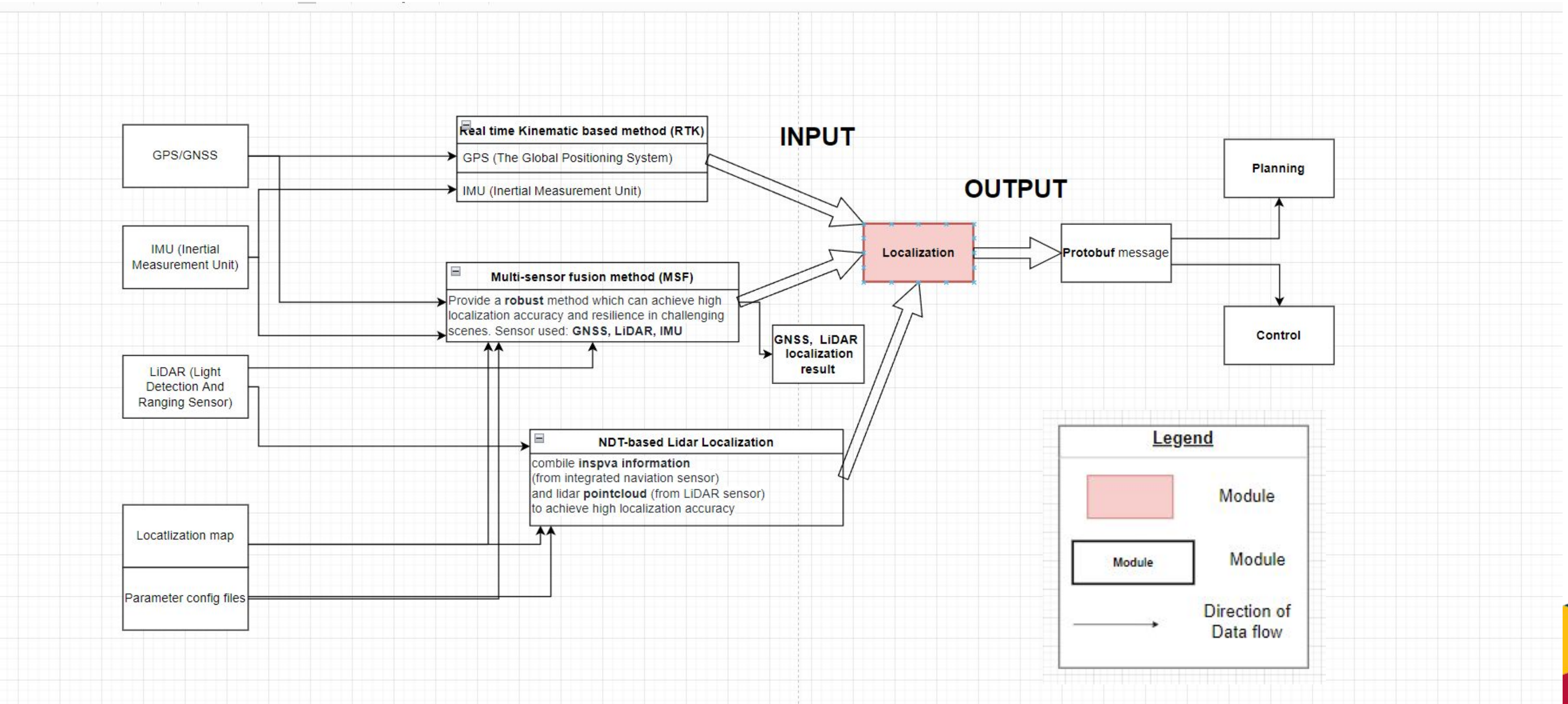
find the shortest route between the departure and destination.

Inputs:

- request of routing map info
- trajectory of the shortest route

Localization

locate the temporal position of the vehicle an out



GPS/GNSS
IMU (Inertial Measurement Unit)
LiDAR (Light Detection And Ranging Sensor)
Locatlization map
Parameter config files
Real time Kinematic based method (RTK)
GPS (The Global Positioning System)
IMU (Inertial Measurement Unit)
Multi-sensor fusion method (MSF)
GNSS, LiDAR localization result
NDT-based Lidar Localization
INPUT
OUTPUT
Localization
Protobuf message
Planning
Control
Legend
Module
Module
Module
Direction of Data flow

Provide a **robust** method which can achieve high localization accuracy and resilience in challenging scenes. Sensor used: **GNSS, LiDAR, IMU**

combile **inspva information**
(from integrated naviation sensor)
and lidar **pointcloud** (from LiDAR sensor)
to achieve high localization accuracy

Perception

recognize surrounding obstacles

Prediction

predict the movement of obstacles and provide each trajectory a proability value

Planning

plans a suitable trajectory of the vehicle, generates a navigation routing.

Control

commands the vehicle, gives orders including acceleration, speed, and steering

CANBus

- CAN(Controller Area Network)
- Bus
- Chassis

Guardian

The final protection

Monitor

Monitoring the entire system

- Hardware monitor
- Software monitor

HMI (DreamView)

The web app that visuallize the current state of the vehicle and driving.

Diagram (Urban Mode)

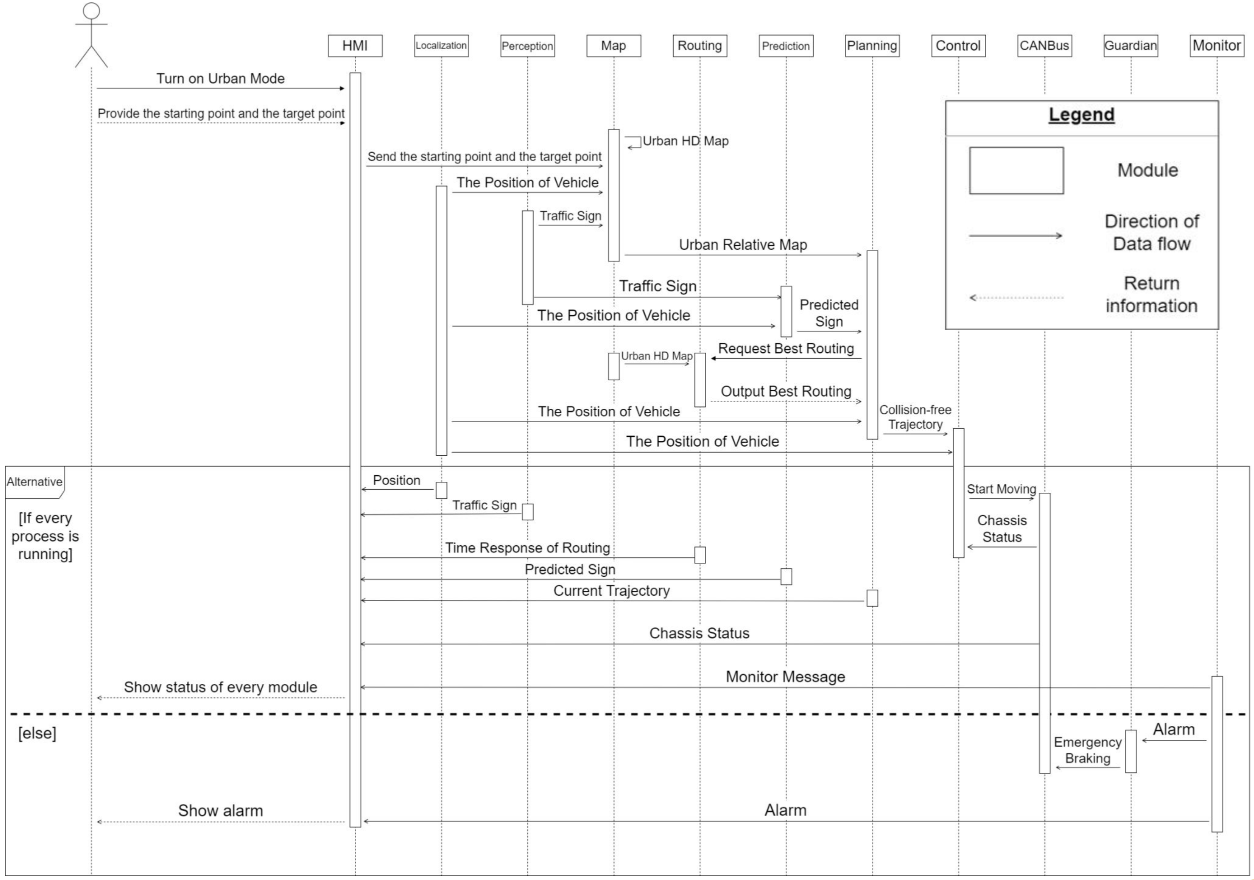
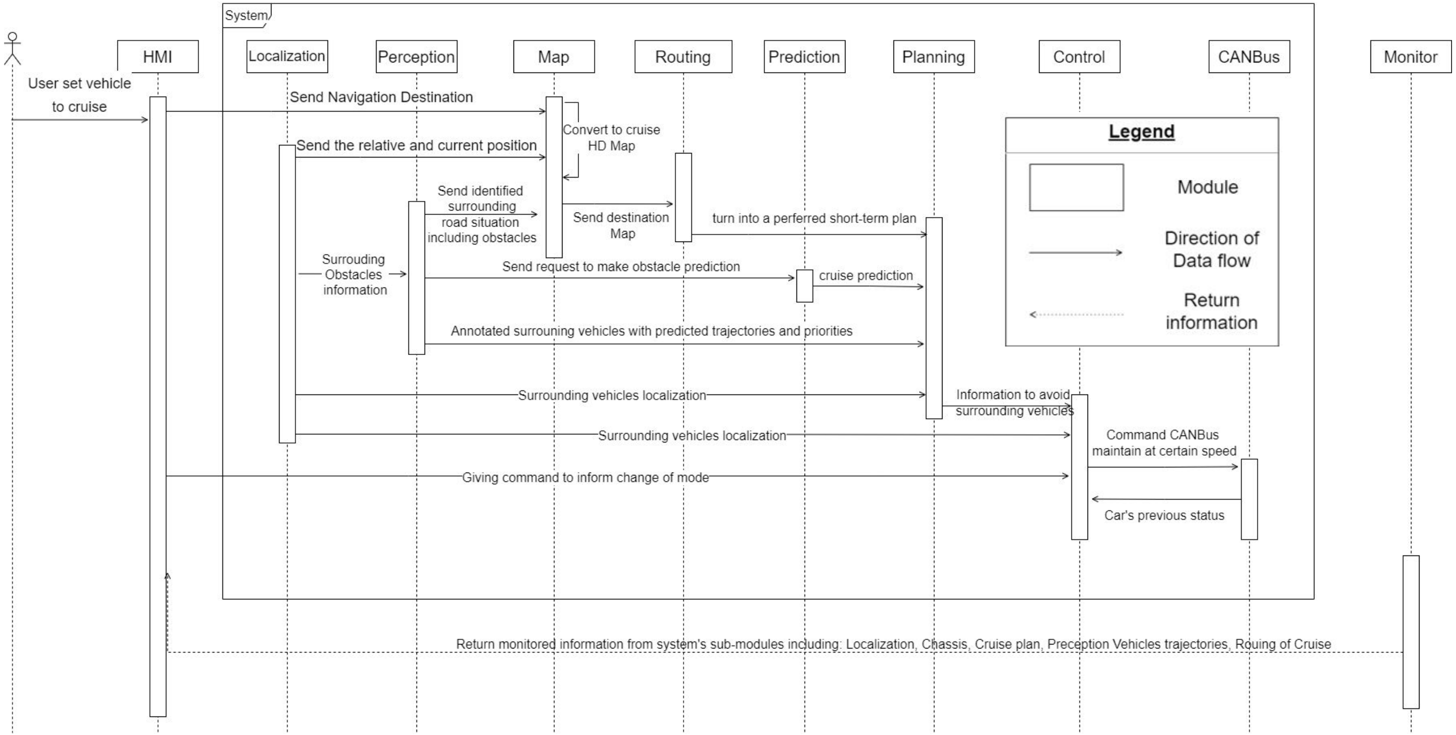


Diagram (Cruise Mode)

Cruise Mode



System

HMI

Localization

Perception

Map

Routing

Prediction

Planning

Control

CANBus

Monitor

User set vehicle to cruise

Send Navigation Destination

Convert to cruise HD Map

Send the relative and current position

Send identified surrounding road situation including obstacles

Send destination Map

turn into a perferred short-term plan

Surrouding Obstacles information

Send request to make obstacle prediction

cruise prediction

Annotated surrouning vehicles with predicted trajectories and priorities

Surrounding vehicles localization

Information to avoid surrounding vehicles

Surrounding vehicles localization

Command CANBus maintain at certain speed

Giving command to inform change of mode

Car's previous status

Return monitored information from system's sub-modules including: Localization, Chassis, Cruise plan, Preception Vehicles trajectories, Rouing of Cruise

Legend

Module

Direction of Data flow

Return information

Derivation

We met problems like ambiguous documents.

Learned Lesson

We understand how software is built up piece by piece and how each single module plays their role in the system.

Conclusion

Pipe-and-filter architecture style.

The whole sub-modules in Apollo are able to cooperate with each other and perform autonomous driving.

Concurrency: highly automatic and quick responses

Reference

apollo developers‘s centre. Apollo. (n.d.). Retrieved February 19, 2022, from https://apollo.auto/developer/index_cn.html#/

Thanks for watching