

The CajunBot Task Planner Architecture for the Urban Challenge



2007 DARPA Urban Challenge (Workshop Presentation)

By

Suresh Golconda

Team CajunBot

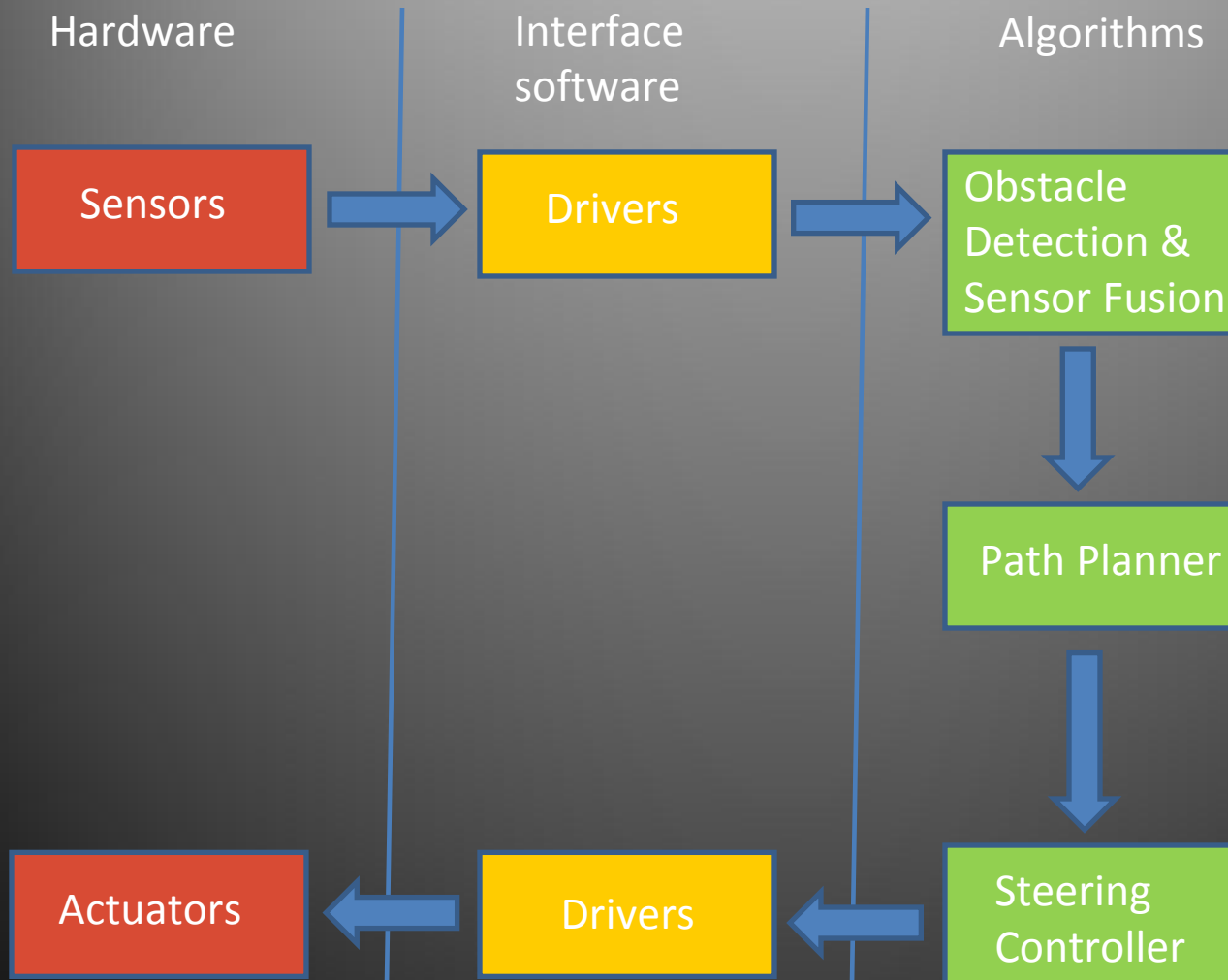
University of Louisiana at Lafayette

www.cajunbot.com

Presentation Overview

- Overall Software Architecture
- Problem Analysis & Basic Idea
- Basic Example Scenario
- Advanced Example Scenario
- Advantages of TP Architecture
- Conclusion & Question

Autonomous Vehicle's Software System



Problem Analysis & Basic Idea

- Requires diverse capabilities

{follow lanes, change lanes, uturn, park, unpark, follow traffic rules, etc}

- Independent capabilities

Task Planner: Intelligent agent specialized in handling specific scenario.




Like follow lane task planner, parking task planner

- Requires modification of the plan based on new situations

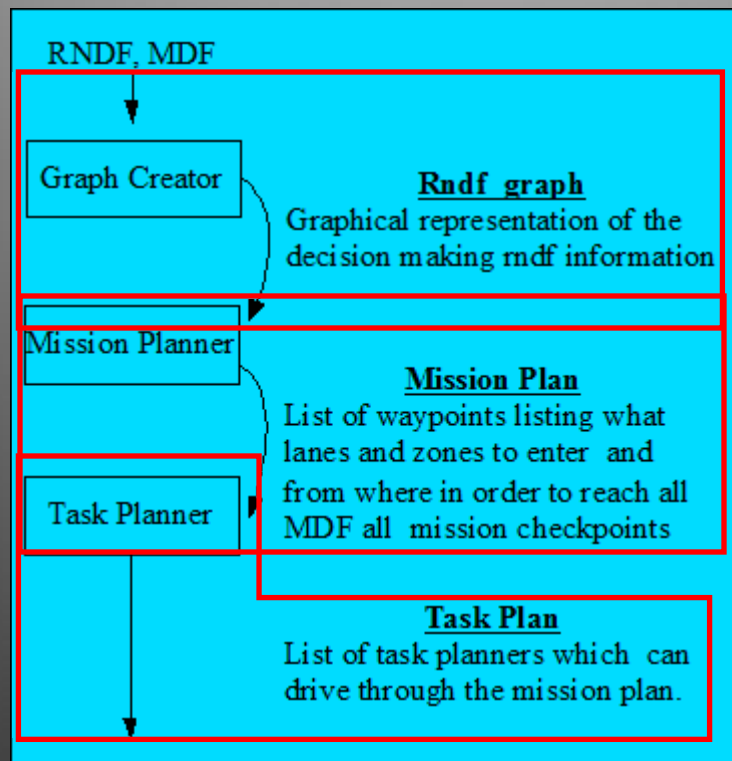
Supervisor: Plan initial system and handle real time situations by modifying 'Task Planner'

Task planners

Task planner	Acronym	Description
Follow lane	FL_TP	Follow the center of the lane from one waypoint to another or for certain distance
Change lane	CL_TP	Generate a smooth path to change from one lane to its neighboring lane
Traffic lane	TL_TP	Generate a smooth path to drive over blocked lane using an on-coming traffic lane
Uturn	UT_TP	Generate a 3 point uturn and staying within the road
Intersection	IT_TP	Generate a curved path through intersection and worry about stop sign, traffic, right of way, etc.
Parking	PK__TP	<i>Generate a path to park the vehicle in a specified spot at specified orientation</i>
Unparking	UP_TP	Generate a path to pull out of a parking spot
Zone navigator	ZN_TP	Generate a path to drive through an open area avoiding obstacles.

-  On road task planners
-  Transition task planner
-  Within zone task planners

Supervisor: Task Plan



Example: Plan modification

Initial Task Plan

FL_TP

New Task

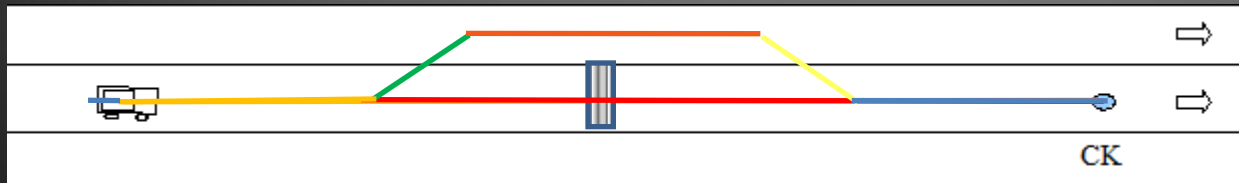
Plan FL_TP

CL_TP

FL_TP

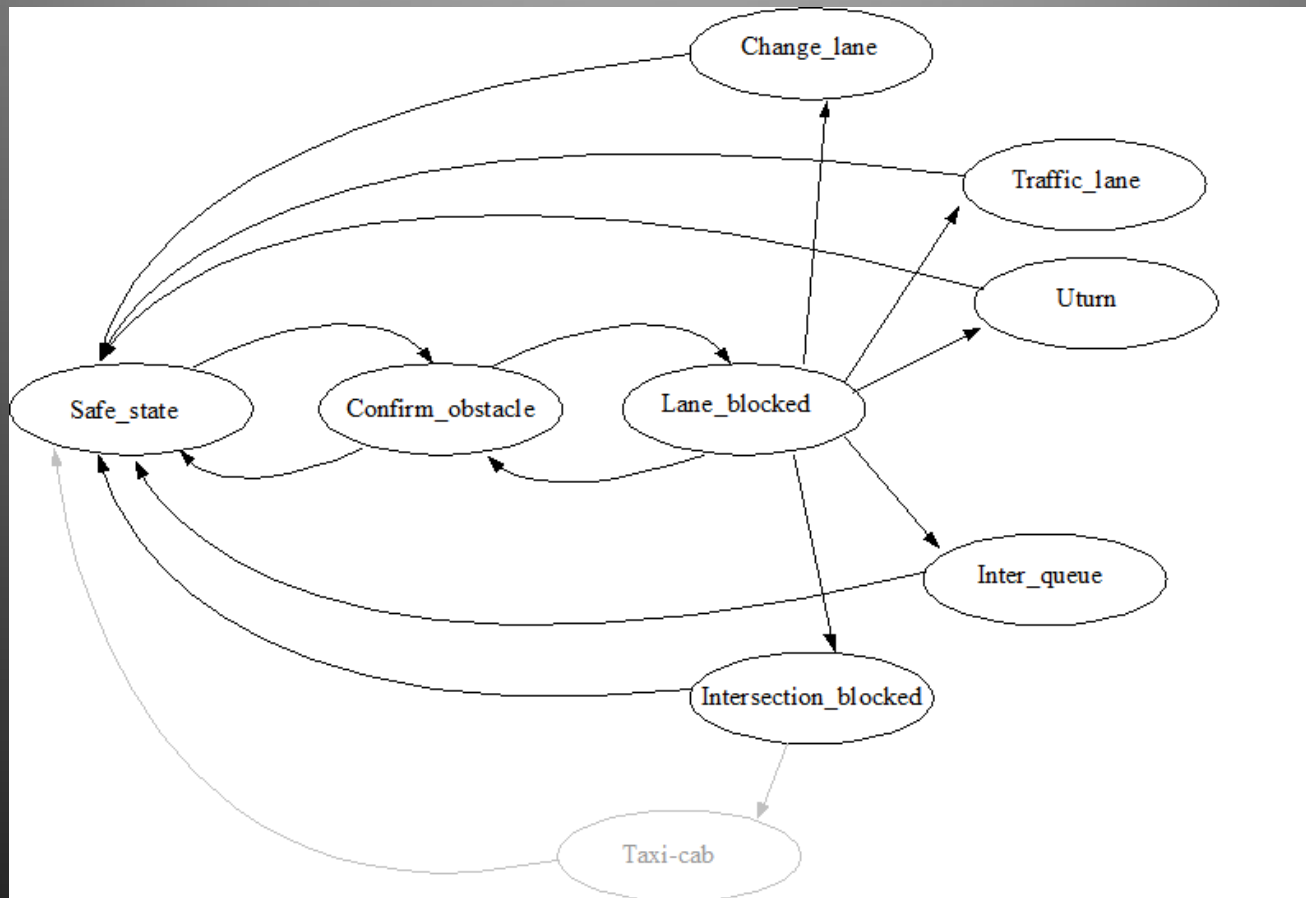
CL_TP

FL_TP



Supervisor Modify Plan

State machine when following lane



Advantages of TP Architecture

- Simplified problem statement
- Incremental development & independent upgrades
- Easy handling of real-time situation encountered (by making decision in terms of task planners)
- Task planner list can guide obstacle detection where to concentration
- Easy modification of path planner's behavior

Possible Issues with TP Architecture

- TP Communication
- Incompatible TPs
- Incompatible TP paths
 - Path direction {forward, reverse}
 - Path tightness {tight path, regular path}

Conclusion

Question?

Questions?

Path Planner Hierarchy

Two level path planner

- Task Planner Handler
 - Determine task planners
 - Maintain & use task planners
 - Modify task planners for situations encountered
- Task Planner
 - Generate path

Path Planner

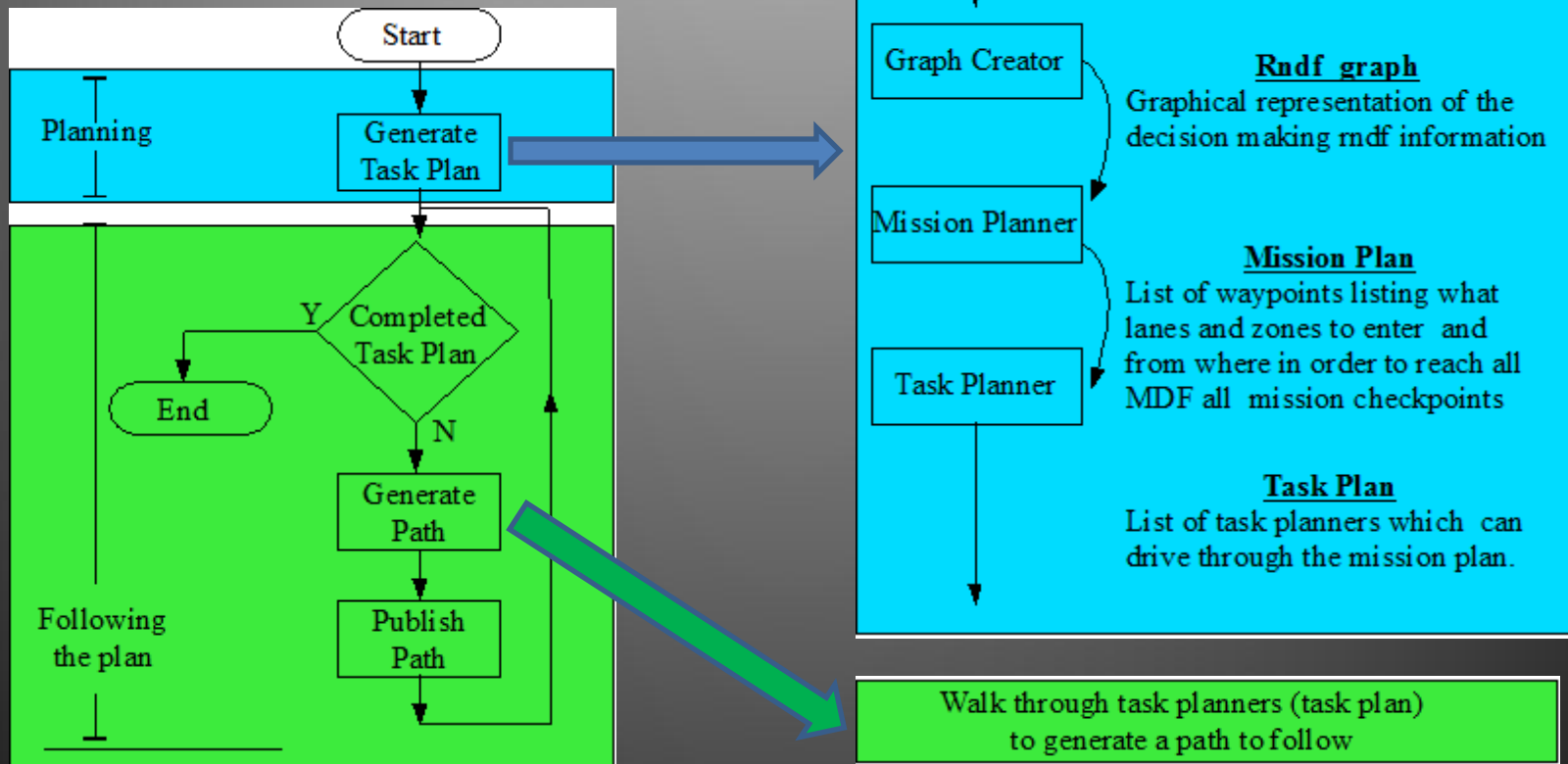
- **Basic Path Planner:**

Driving through urban world with no traffic or lane blockages

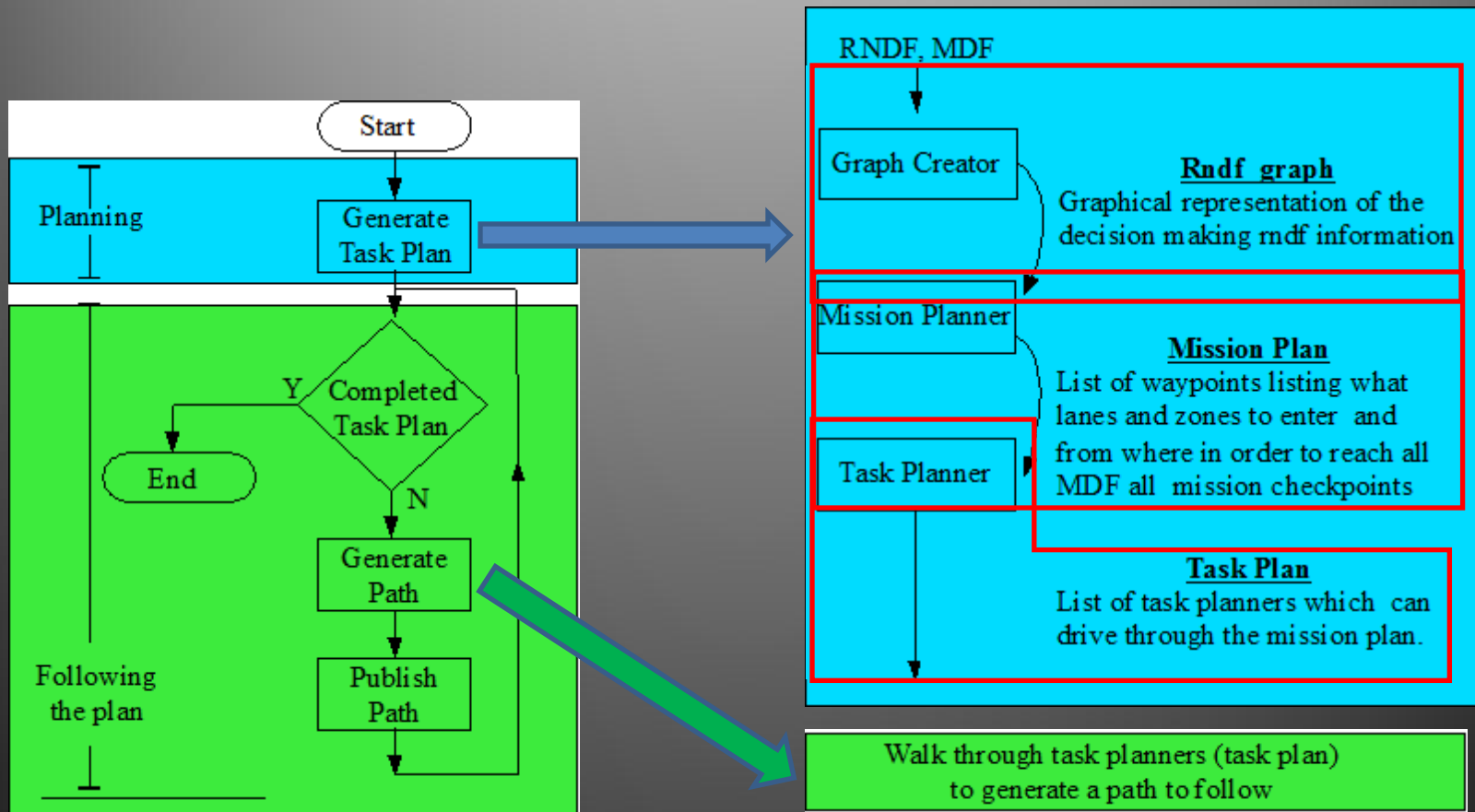
- **Advanced Path Planner:**

Driving lane blockages and traffic

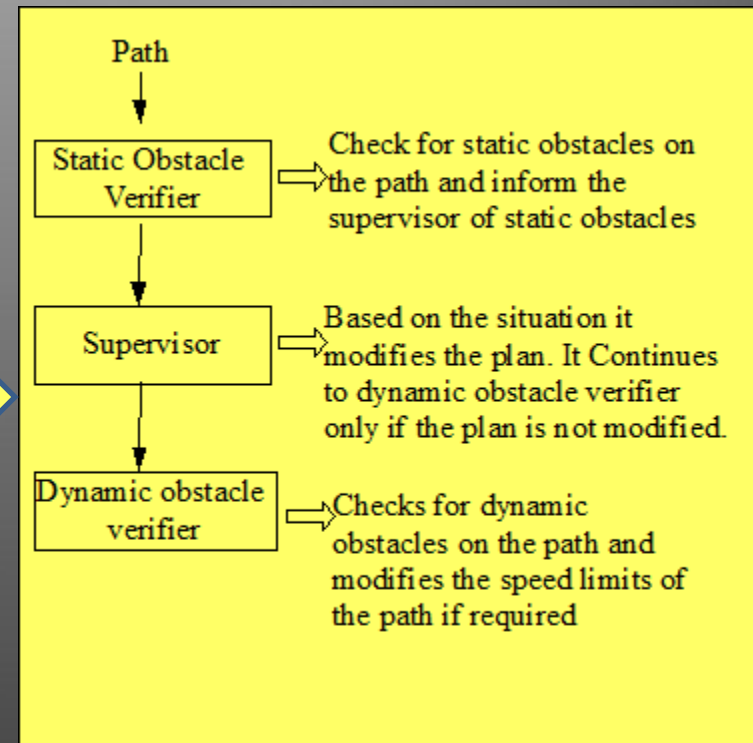
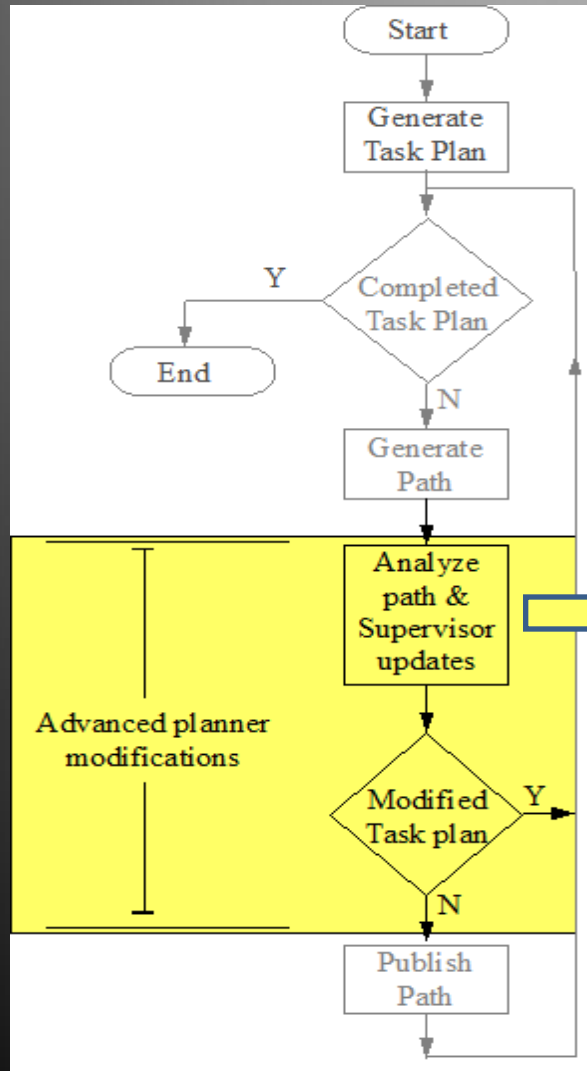
Basic Path Planner Architecture



Basic Path Planner Architecture



Advanced Path Planner



Future Upgrades

- Ability to backup for new plan
- Take dynamic traffic more serious
- Look for dynamic traffic off the lane
- Supervisor to handle situations that 'zone task planner' cannot handle locally

Basic Idea

Develop intelligent agents each of which are specialized in handling a specific scenario, like follow lane, change lane, etc. (*Task Planner*)

Perform the planning of mission and handling of new situations encountered by using these Task Planners (**Task Plannner Handler**)