Software Requirements Specification (SRS)
Cooperative Adaptive Cruise Control 3

Authors: Audrey Guest, Josh Jarvis, Manish Rajendran, Danielle Kelley, Jack Brooks
Customer: Mr. William Milam, Ford Motor Company
Instructor: Dr. Betty Cheng

1. Introduction

1.1. Purpose

The purpose of this document is to offer a detailed description of the Cooperative Adaptive Cruise Control system. The document will explain the features of the system and their purpose, the system's interface, how the system will function, how the system reacts to the external environment, and the constraints under which it must operate. This information is intended for the stakeholders and development team.

1.2. Scope

The Cooperative Adaptive Cruise Control (CACC) system is a software-based electronic control system that increases the convenience and safety features provided to drivers. The software of the system will utilize a combination of information from the cameras, sensors, and radio communications to provide an interface for the user's dashboard. From the dashboard the user will be able to enable or disable the system, and view warnings as they appear.

Once enabled, the system will maintain a constant vehicle speed. The system is designed to set up a platoon of vehicles that follow and collect speed and location information from a lead vehicle, while maintaining a safe following distance. If a driver wishes to leave the platoon, they can disable the system and the system will notify the appropriate vehicles in the platoon.

1.3. Definitions, acronyms, and abbreviations

Term	Definition
CACC++, CACC	Cooperative Adaptive Cruise Control
Following vehicle	The vehicle that is directly behind a vehicle in a platoon.
Platoon	A grouping of vehicles, following one after another at a constant speed.
Software Requirements Specification (SRS)	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
Target vehicle	The vehicle that is used as the point of reference for speed for the vehicle behind it.
V2V	Vehicle to vehicle communication. V2V is the way in which vehicles can wirelessly transit information to each other.

1.4. Organization

Section 2 of the SRS document provides an overview of the CACC system's functionality and its constraints. Section 2.1 specifically describes the context and constraints of the system. Section 2.2 describes the functionality of the CACC system as a whole and how the system will perform. Section 2.3 explains what is expected from the user in order for the CACC system to function correctly. Section 2.4 describes the constraints of the system and section 2.5 explains the assumptions and dependencies. Finally, section 2.6 explains requirements that are out of the scope of the current product, but could be implemented in the future.

Section 3 describes the specific requirements of the CACC system and provides further requirements relating to the security of the CACC system.

Section 4 of the SRS document shows and describes use case diagrams, domain models, sequence diagrams, and state diagrams that show the specific scenarios of the CACC product and how their behavior would be handled.

The fifth section of the SRS document offers information about the prototype, how it works and how to access it. Section 5 specifically covers the functionality of the prototype. Section 5.1 explains what is needed to run the prototype and section 5.2 gives a sample scenario of the CACC system.

Finally, section 6 of the document enumerates the resources used to create the requirements of the project and section 7 is a point of contact for the project.

2. Overall Description

This section will provide background information for the CACC system/product. A broad view of how the product fits with other systems will be provided, along with the main purposes of the product and its constraints. Expectations for the user of CACC will be laid out in order to give further context for the product. Any assumptions regarding hardware, software, the system environment, user interactions, or other factors will be noted. Finally, requirements that are determined to be beyond the scope of our project, but reasonable for future versions will conclude this section.

2.1. Product Perspective

CACC is one of many subsystems that comprise the larger system of a vehicle. It is not essential, and in fact few vehicles currently possess it as part of their systems. However, it is reasonable to project that CACC will become increasingly popular and eventually an essential element of a vehicle in order to increase safety and autonomy for the driver. Figure 1 shows the relationship of the CACC system to a vehicle in relation to other essential parts of a vehicle. The essential components of the vehicle include the engine, brakes, accelerator, dashboard, and steering wheel. The vehicle must have one of each of these in order for it to operate, and they can all exist as parts without the vehicle. CACC, however, is a nonessential part of the vehicle that cannot exist on its own without being connected to a vehicle. CACC is a system that will assist the larger system of a vehicle.

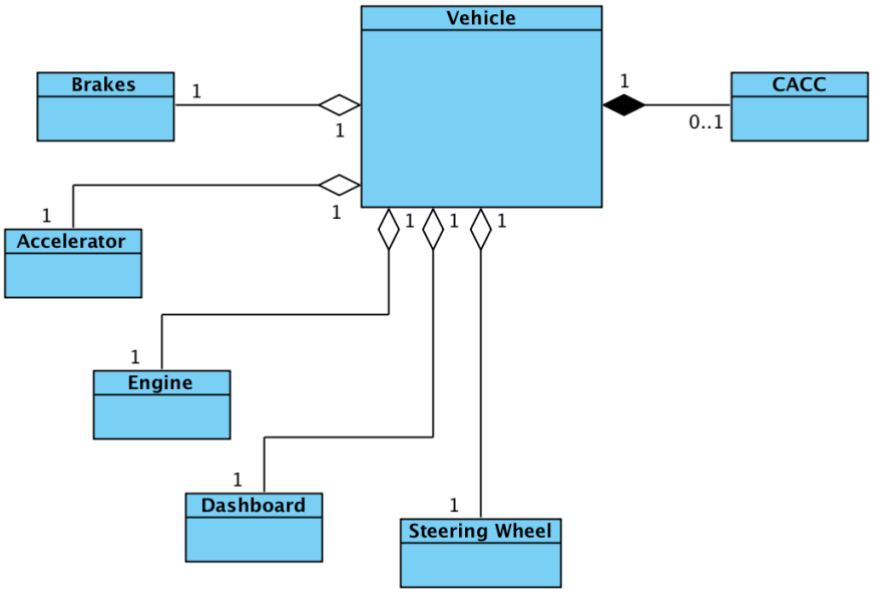

Figure 1: CACC's relationship to a larger vehicle system.

The CACC system has both interface and memory related constraints. The environment system constraints include weather and road conditions that could affect the abilities of the camera and radar sensors. The weather and road condition constraints can also be seen as hardware constraints. Rainy or snowy weather can cause the camera or radar sensors to fault. Another hardware constraint is that each vehicle with CACC must have a dashboard for a user interface and hardware for radio communication. Software constraints include having to choose when one feature of CACC must override another feature in order to keep the driver safe. Software constraints also include the large amount of memory needed for the system because CACC holds data for its own vehicle as well as target and following vehicles. A final constraint is for CACC to be fully operational and work with platoons of cars, the majority of cars on the road must also be equipped with CACC, or some sort of V2V technology, to allow the vehicles to communicate with each other.

2.2. Product Functions

The CACC system's main functions are that it provides a more effective adaptive cruise control when the vehicle is in a platoon, or group of other CACC vehicles. If the vehicle is the lead vehicle in the platoon, the system will act as a normal Adaptive Cruise Control system adjusting

speed based on the non CACC vehicle in front of it. The lead vehicle will also communicate with the platoon vehicles behind it in order to minimize following distance. If the vehicle is in a platoon and is not the lead, it will communicate with the vehicle in front of it in order to receive data about the vehicle. The vehicle will be constantly calculating the safest following distance while changing speed in order to maintain that dynamic distance. If the vehicle is not in a platoon at all, it will act as a normal Adaptive Cruise Control system with the exception that the vehicle will be trying to find a platoon to join.

2.3. User Characteristics

It is expected that the driver is licensed to drive. Due to the extensive capabilities of CACC, the driver can only enable/disable the system with a push of a button. The driver should preferably have general knowledge of CACC so that they are not surprised by its behavior. The primary responsibility of the driver in the system is simply observing the CACC software guide the vehicle in different scenarios.

2.4. Constraints

There are features of the vehicle that must be operational in order for the system to be enabled, disabled or to operate properly. It is critical that the acceleration, deceleration and steering wheel in the vehicle are functional in order for the CACC system to operate. The system will also be inoperable if there is a delay in radio communication between the target vehicle and the following vehicle. This communication is critical in the maintenance of the appropriate speed and can affect the overall safety of the system when enabled. Furthermore, if there is a delay in communication between a target vehicle and its following vehicle, the following vehicle will not know what speed or location to adjust itself to. This lack of knowledge can cause the safe distance to be violated and possibly cause a collision.

2.5. Assumptions and Dependencies

We are operating under various assumptions regarding the CACC system in order to accurately assess its behavior. One assumption about the software is that it is much more effective at guiding a vehicle than a human driver. For this reason, the driver should have limited override

capability. In case of emergency, the driver can revert back to ACC. Our assumption regarding the environment is that the CACC system will be able to adjust to extreme weather conditions. So long as conditions are not deemed so dangerous that driving is not advised, the system should be able to utilize weather data to guide the vehicle. Finally, we assume that nearly all vehicles on the road have CACC or some sort of system that allows them to communicate with other vehicles. The primary benefit of CACC over ACC is communication between vehicles, and this is rendered ineffective if there are a significant amount of vehicles without the software required.

2.6. Apportioning of Requirements

Some initial requirements of the CACC system have been determined to be out of the current scope of the project and may be addressed in future versions of the product. An enumerated list of the future requirements are listed below.

1. A platoon is capable of splitting into two platoons the account for an individual car entering or leaving the platoon.
2. CACC will control the steering wheel in order to change lanes for object avoidance.
3. There will be a button for the driver to press to autonomously change lanes.
4. CACC will use GPS to determine traffic data and autonomously choose the best route for the vehicle to take based on the data.
5. The user interface for CACC will include vehicle diagnostics.

3. Specific Requirements

1. The vehicle will have a front camera to detect and calculate the speed of the vehicle in front of it (target vehicle).
    1.1. The camera will identify the target vehicle and find its speed and location.
    1.2. The camera will act as a backup for the radar sensors if there is a failure.
2. The vehicle will have radar sensors to detect and find the speed of the vehicle in front of it.
    2.1. The radar sensors will detect, identify, and track the vehicle ahead of it.
3. Append sensors to appropriate place on the vehicle. Optimal placement is that which will allow full awareness of objects in front of vehicle.
4. Sensors must be able to detect objects up to 300 meters in front of the vehicle and 150 meters behind the vehicle.

5. The vehicle will be capable of being a part of a platoon, or a large line of vehicles all travelling at the same speed.
    5.1. Communicate with platoon leader/platoon member ahead of vehicle to adjust speed accordingly.
    5.2. V2V hardware must be included in the hardware stack. This may include antennas, GPS, network cards, or multiple embedded systems.
    5.3. The vehicle will use GPS to communicate with the other cars in the platoon.
        5.3.1. The GPS will track the location, speed, and direction of the other vehicles in the platoon.
        5.3.2. The GPS will help the radar sensor detect the difference between target cars, and fixed objects.
        5.3.3. The GPS will maintain appropriate distance between the vehicles in the platoon if the radar fails.
    5.4. Vehicles will receive information about braking and accelerating capabilities of the target vehicle.
        5.4.1. Constraint: Vehicles can only decelerate at 2 Gravity and vehicles have different braking and accelerating capabilities based on their size.[1]
6. There must be software that can detect when one feature must override another. For example, a speed/hill warning that causes deceleration should override the adaptive cruise control that may want acceleration.[1]
7. The vehicle will use radio communication to communicate with other cars in the platoon.
    7.1. The radio system can be restarted by the vehicle controller.
8. There will be an electric throttle that can control the cars speed by adding or removing power.
9. The vehicle will brake by wire, or apply the brakes, to regulate the vehicles speed.
10. There will be a vehicle controller that connects all of the sensors and systems together.
    10.1. The vehicle controller will take in the speed of the vehicle and target vehicle and adjust the vehicle's speed accordingly.
    10.2. The vehicle controller will command the throttle and brakes.
    10.3. The vehicle controller will receive information from the radar sensors and camera.
    10.4. The vehicle controller will communicate with the radio communication system.
11. There will be a system to detect a failure to start a task and detect memory leaks.
    11.1. The system will decide how to recover from these memory leaks.
12. There will be an independent monitoring function to ensure that the values being sent to the throttle match the actual input values from the target vehicle.
13. There will be additional safety features to assist the Cooperative Adaptive Cruise Control (CACC) system. The safety features must not cause any software faults for the system.
    13.1. There will be adaptive cruise control.
    13.2. There will be lane keeping / lane centering.
    13.3. There will be curve speed assist.

13.4. There will be hill management.

14. The CACC system must work with congested traffic, various weather and road conditions, various states of vehicle health, and varying skill levels of drivers.[1]
    14.1. Prompt user if one or more sensors are obstructed.
    14.2. Notify audibly and visually that the system is unavailable.
    14.3. Will function regardless of varying road slope/grade.
15. The system must have a simple interface for the user.
16. System must be enabled by default.
    16.1. The system must have override available to the user in case of malfunction.
17. Must maintain speed and gap setting set by user.
18. There must be an automatic procedure to join and leave the platoon regardless of location in the platoon due to safety reasons.
19. If CACC becomes unavailable, the system will fall back to ACC. If ACC is unavailable, the system will fall back to regular cruise control.
20. There will be system status icons for CACC and ACC displayed all the time.
21. Secure system to prevent malicious attacks.
22. Alert driver of potential hazards/collisions.

Cyber Security Requirements:

23. CACC system must constantly check whether vehicle in platoon (most importantly the lead vehicle of a platoon) has been compromised.
24. Messages that are sent to other vehicles must be encrypted.
25. Messages that are received from other vehicles must be able to be decrypted.
26. Messages that are received from other vehicles must be checked to ensure integrity.
27. Synchronized clocks must be used to protect against replay attacks (data fraudulently repeated or delayed).
28. Digital signatures must be used to counteract message falsification attacks.
29. Must search for inconsistencies between radar and camera and fall back to ACC if any are found.
30. System must collect messages between vehicles in platoon and check them against each other to determine whether a jamming attack (transmission of interfering radio signals) has occurred.
    30.1. All vehicles in platoon should fall back to ACC.

4. Modeling Requirements

Use Case Diagram:

System Boundary: The system boundary is the sensors, camera, and everything within and including the CACC system.

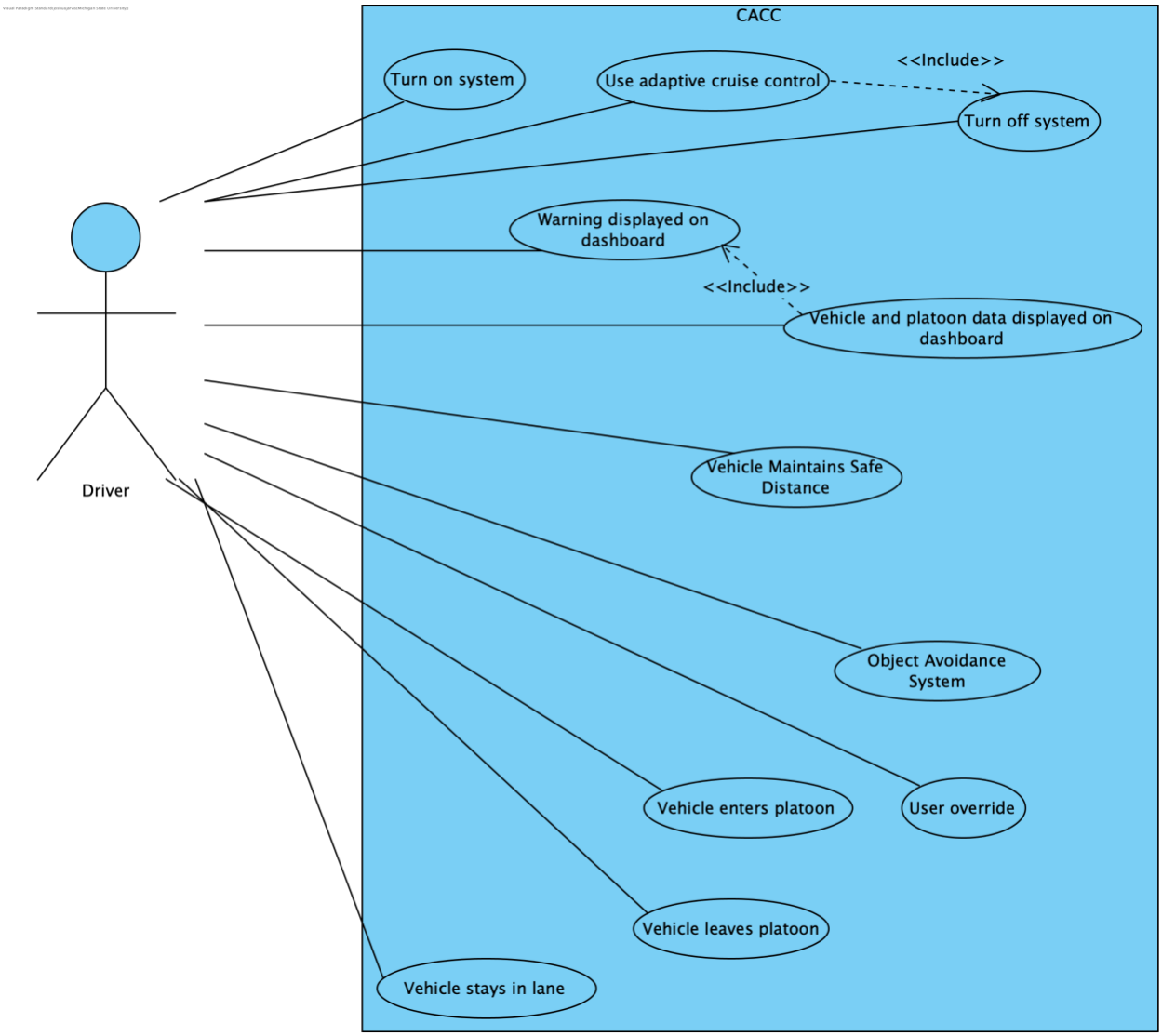

Figure 2: Use case diagram.

Use Case Documentation:

Use Case:	Object Avoidance System
Actors:	Driver
Description:	Vehicle radar sensors or camera detects an object, and sends the data to the vehicle controller. The vehicle controller communicates to the other cars in the platoon using vehicle to vehicle communication, causing all vehicles in platoon to adjust accordingly. Adjustment includes change of speed and/or change of direction.
Type:	Primary

Includes:	N/A
Extends:	N/A
Cross-refs:	#1, 1.1, 1.2, 2, 2.1, 3, 4, 5, 5.1, 5.2, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.4, 5.4.1, 7, 7.1, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 14, 21
Use cases:	N/A

Use Case:	Turn off system
Actors:	Driver
Description:	If driver would like to turn off the system and regain control of the vehicle, the driver can hit a button on the user interface. The vehicle will leave a platoon if it is in one and transfer vehicle control to the driver. The system will go from CACC to ACC (Adaptive cruise control).
Type:	Primary
Includes:	Use adaptive cruise control
Extends:	N/A
Cross-refs:	13, 13.1, 14, 16, 16.1, 19, 21
Use cases:	N/A

Use Case:	Turn on system
Actors:	Driver
Description:	In order to enable the system, the driver must push a button on the user interface. This will cause the vehicle to start driving autonomously and begin searching for a platoon or another vehicle to create a platoon.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	#1, 1.1, 1.2, 2, 2.1, 3, 4, 5, 5.1, 5.3, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.4, 5.4.1, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 14, 21, 23, 24, 25, 26, 27, 28, 29, 30, 30.1

Use cases:	N/A

Use Case:	Use adaptive cruise control
Actors:	Driver
Description:	The vehicle will fallback onto adaptive cruise control when the system's cooperative component becomes unavailable. When the vehicle is using adaptive cruise control, it will not be able to join a platoon nor will it be able to maintain a smaller gap.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	#1, 1.1, 1.2, 2, 2.1, 3, 4, 8, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 13, 13.1, 14, 17, 19, 21
Use cases:	Turn off system

Use Case:	User override
Actors:	Driver
Description:	While in the platoon, the CACC system will have control of the vehicle. Any driver input in relation to speed will be ignored (i.e. the driver hitting the brakes will not cause the vehicle to brake). The driver can turn off the system to override and regain control.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	6, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 14, 16, 16.1, 21
Use cases:	Vehicle enters platoon

Use Case:	Vehicle and platoon data displayed on dashboard

Actors:	Driver
Description:	On the vehicle's dashboard, the vehicle's current speed and the target vehicle's speed, acceleration, and deceleration will be available to the driver through radio, GPS, camera and sensor detection. Once the vehicle leaves the platoon, the data will no longer be available for the driver to access.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	#1, 1.1, 1.2, 2, 2.1, 3, 4, 5, 5.1, 5.2, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.4, 5.4.1, 7, 7.1, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 14, 15, 20, 21, 23, 24, 25, 26
Use cases:	Warning displayed on dashboard

Use Case:	Vehicle enters platoon
Actors:	Driver
Description:	When the system is activated, the vehicle searches for a platoon of vehicles using vehicle-to-vehicle communication. When a platoon is found, the system will automatically join the platoon by getting behind the last vehicle in the platoon and making it the target vehicle. It will then sync the speed of all of the vehicles in the platoon.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	5, 5.1, 5.2, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.4, 5.4.1, 7, 7.1, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 14, 17, 18, 21, 23, 24, 25, 26, 27, 28, 29, 30, 30.1
Use cases:	N/A

Use Case:	Vehicle leaves platoon
Actors:	Driver

Description:	If the driver decides to leave the platoon they will initiate the leave platoon function. Similarly, if a fault occurs within the system, the leave platoon function will be initiated and the user will be warned that the system is turning off. If the vehicle that wishes to leave the platoon is the lead vehicle or a vehicle with another vehicle in front or behind it, it will send out a message to the vehicle behind it. If the vehicle that wishes to leave the platoon is the last vehicle it will decelerate.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	5, 5.1, 5.3, 5.3, 5.3.1, 5.3.3, 5.4, 5.4.1, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 14, 18, 21
Use cases:	N/A

Use Case:	Vehicle Maintains Safe Distance
Actors:	Driver
Description:	The system detects an object in front of the vehicle by using radar, sensors, and GPS. The system will then use vehicle-to-vehicle communication with the target vehicle in front of it. The vehicle controller will receive data from the target vehicle and send data to the vehicle behind it and compute whether the vehicle needs to accelerate, decelerate, or stay the same speed in order to maintain the correct distance from the vehicle in front of it. The car may need to adjust its speed when travelling around a curve or down a hill. The vehicle may need to accelerate when travelling up a hill.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	#1, 1.1, 1.2, 2, 2.1, 3, 4, 5, 5.1, 5.2, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.4, 5.4.1, 6, 7, 7.1, 8, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 13, 13.4, 14, 17, 21
Use cases:	N/A

Use Case:	Vehicle stays in lane

Actors:	Driver
Description:	The system will communicate with the vehicle's lane management system in order to autonomously stay within the lane that the vehicle is in. It will still be syncing with the target vehicle in the platoon to ensure they are the same speed.
Type:	Primary
Includes:	N/A
Extends:	N/A
Cross-refs:	#1, 1.1, 1.2, 2, 2.1, 3, 4, 6, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 13, 13.2, 14, 21
Use cases:	N/A

Use Case:	Warning displayed on dashboard
Actors:	Driver
Description:	Occurs when there are any components unoperational within the system. There will be an indication on the vehicle's dashboard. This can include sensor obstructed warning, potential hazard warning, or CACC system unavailable warning.
Type:	Primary
Includes:	Vehicle and platoon data displayed on dashboard
Extends:	N/A
Cross-refs:	#1, 1.1, 1.2, 2, 2.1, 3, 4, 5, 5.1, 5.2, 5.3, 5.3.1, 5.3.2, 5.3.3, 5.4, 5.4.1, 7, 7.1, 10, 10.1, 10.2, 10.3, 10.4, 11, 11.1, 14, 14.1, 14.2, 14.3, 15, 21, 22
Use cases:	Vehicle and platoon data displayed on dashboard

Domain Model:

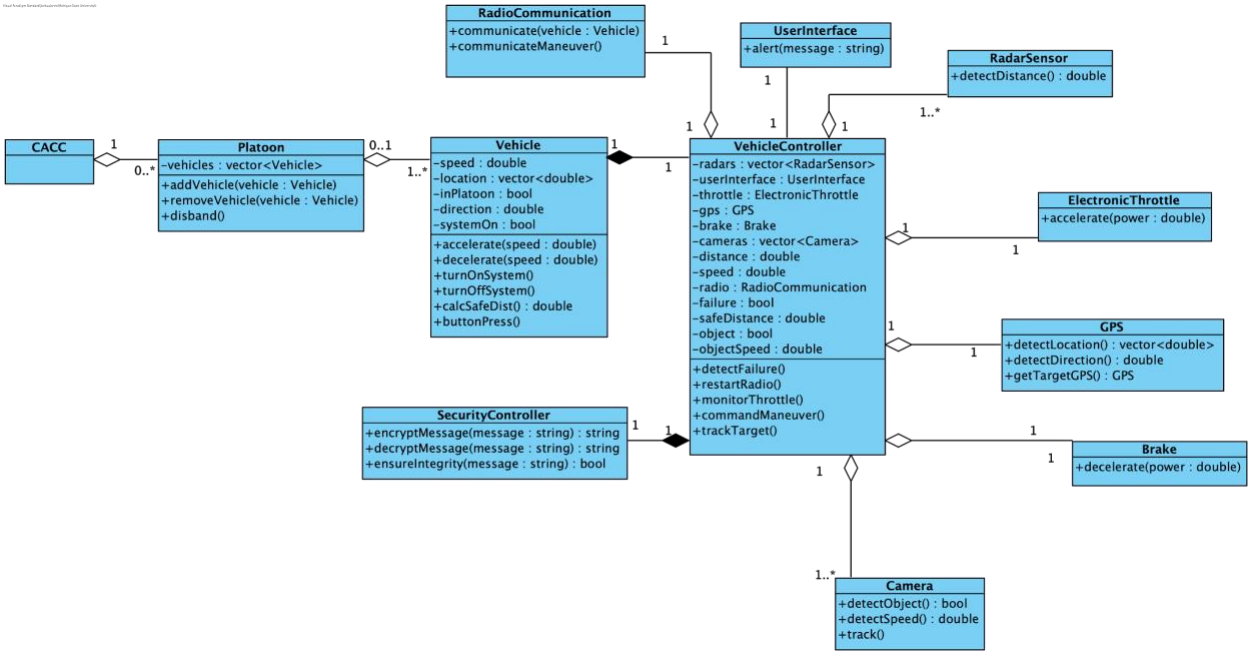

Figure 3: Domain model.

Data Dictionary for Domain Model:

Element Name		**Description**
Brake		Brake represents the brakes of a vehicle.
Operations		
	decelerate(power : double): void	Apply the given power to the brakes by wire to decelerate the vehicle.
Relationships	Brake is a part of VehicleController. It can be commanded by VehicleController to decelerate the vehicle by a certain speed.	
UML Extensions	Brake is an aggregation of VehicleController. Brake can have one VehicleController.	

Element Name		**Description**
CACC		It encapsulates the entire system that is used by platoons and cars.
Relationships	CACC is used by all platoons and any vehicle that is equipped with the CACC system. CACC represents the system as a whole.	
UML Extensions	Platoon is an aggregation of CACC. CACC can have 0 to infinite Platoons.	

Element Name		**Description**
Camera		Class that represents a camera that detects objects in front of the vehicle
Operations		
	detectObject() : bool	Returns true if an object is in front of the vehicle
	detectSpeed() : double	Detects and returns the speed of the object ahead.
	track() : void	Track the target vehicle in order to account for a vehicle going around a curve.
Relationships	Camera is a part of VehicleController. It can be commanded by VehicleController to detect an object in front of the vehicle	
UML Extensions	Camera is an aggregation of VehicleController. Camera can have 1 VehicleController	

Element Name		**Description**
ElectronicThrottle		Class that represents the electronic throttle of the vehicle. Used to accelerate the vehicle
Operations		
	accelerate(power : double): void	Accelerates the vehicle by sending the given amount of power to the throttle.
Relationships	ElectronicThrottle is a part of VehicleController. It can be commanded by VehicleController to accelerate the vehicle by a certain speed.	
UML Extensions	ElectronicThrottle is an aggregation of VehicleController. ElectronicThrottle can have one VehicleController.	

Element Name		**Description**
GPS		Class that represents GPS system to detect the location and direction of vehicle
Operations		
	detectLocation(): vector<double>	Returns a double of the longitude and latitude of the vehicle
	detectDirection(): double	Returns a double of the direction of the vehicle
	getTargetGPS() : GPS	Returns the GPS object of the target vehicle in the platoon.
Relationships	GPS is a part of VehicleController. It can be commanded by VehicleController to detect the location of the vehicle and the direction the vehicle is travelling in.	

UML Extensions	GPS is an aggregation of VehicleController. GPS can have one VehicleController.

Element Name		**Description**
Platoon		Class that represents a platoon of vehicles that communicates with each other using the CACC system
Attributes		
	vehicles : vector<Vehicle>	A vector of the vehicles in the platoon.
Operations		
	addVehicle(vehicle : Vehicle): void	Vehicle joins the platoon
	removeVehicle(vehicle : Vehicle): void	Vehicle leaves the platoon
	disband(): void	All vehicles leave the platoon
Relationships	Platoon is a part of CACC. Vehicle is a part of Platoon. Vehicles can join and leave a platoon.	
UML Extensions	Platoon is an aggregation of CACC. Platoons have 1 CACC. Vehicle is an aggregation of Platoon. Platoons are made up of 1 or more Vehicle.	

Element Name		**Description**
RadarSensor		RadarSensor represents each radar sensor used in the CACC system.
Operations		

	detectDistance(): double	This function detects and returns the distance of the target vehicle.
Relationships	RadarSensor is a part of VehicleController. It can be commanded by VehicleController to detect the speed of the vehicle.	
UML Extensions	RadarSensor is an aggregation of VehicleController. RadarSensor can have one VehicleController.	

Element Name		**Description**
RadioCommunication		RadioCommunication represents the radio communication software used by CACC for communication between vehicles.
Operations		
	communicate(vehicle : Vehicle): void	This function communicates, i.e. retrieves data, from a given vehicle.
Relationships	RadioCommunication is a part of VehicleController. It can be commanded by VehicleController to communicate with another given vehicle.	
UML Extensions	RadioCommunication is an aggregation of VehicleController. RadioCommunication can have one VehicleController.	

Element Name		**Description**
SecurityController		SecurityController represents the software used to secure the CACC system from any outside attacks.
Operations		

	encryptMessage(message : string) : string	Encrypt data that will be sent from the vehicle and given to another. Returns the encrypted message.
	decryptMessage(message : string) : string	Decrypt data that has been sent from another vehicle. Returns the decrypted message.
	ensureIntegrity(message : string) : bool	Ensure the integrity of data sent from another vehicle. Return bool whether or not the message is safe.
Relationships	SecurityController is a part of VehicleController. It can be commanded by VehicleController to encrypt the CACC system's software.	
UML Extensions	SecurityController is a composition of VehicleController. SecurityController can have one VehicleController.	

Element Name		**Description**
UserInterface		Class that represents a user interface. The user interface alerts the driver and shows data.
Operations		
	alert(message : string) : void	Warns driver by displaying given message on user message.
Relationships	UserInterface is a part of VehicleController. It can be commanded by VehicleController to alert the driver.	
UML Extensions	VehicleController has an association to UserInterface. UserInterface has 1 VehicleController.	

Element Name		**Description**
Vehicle		Vehicle represents a vehicle that will be equipped with CACC software. It may or may not be part of a platoon.
Attributes		
	speeed : double	The speed of the vehicle.
	location : vector<double>	The location of the vehicle.
	inPlatoon : bool	A bool stating whether or not the Vehicle is in a platoon.
	direction : double	The direction the Vehicle is travelling in.
	decelConstant : double	The decelerating capabilities of the Vehicle.
	accelConstant : double	The accelerating capabilities of the Vehicle.
	systemOn : bool	A bool stating whether or not the system is on. It is True by default.
Operations		
	accelerate(speed : double): void	Accelerate the Vehicle to a certain speed.
	decelerate(speed : double): void	Decelerate the Vehicle to a certain speed.
	turnOnSystem() : void	Command the vehicle controller to turn on the system.
	turnOffSystem() : void	Command the vehicle controller to turn off the system.

	calcSafeDist() : double	Calculate and return the safe stopping distance for the vehicle at the current speed.
	buttonPress() : void	The user has pressed the on or off button on the dashboard.
Relationships	Vehicle is a part of Platoon. It also communicates with VehicleController to get all of the data needed for CACC to function.	
UML Extensions	Vehicle is an aggregation of Platoon. VehicleController is a composition of Vehicle. Vehicle can have zero or one Platoon and one VehicleController.	

Element Name		**Description**
VehicleController		Class that represents the vehicle controller. The vehicle controller controls all sensors and communication links for the CACC system.
Attributes		
	radars : vector<RadarSensor>	A vector of all the radar sensors in the system.
	userInterface : UserInterface	The user interface for the vehicle.
	throttle : ElectronicThrottle	The vehicle’s electronic throttle.
	gps : GPS	The vehicle's GPS system.
	brake : Brake	The vehicle’s brakes.
	cameras : vector<camera>	A vector of all the cameras used in the system.
	speed : double	The speed of the vehicle that the vehicle controller is a part of.

	radio : RadioCommunication	The vehicle's radio communication object.
	failure : bool	True if there was a failure to start task or a memory leak.
	safeDistance : double	The distance that the vehicle should be behind the target vehicle.
	object : bool	True if there is an object that is detected.
	objectSpeed : double	The speed of the object in front of the vehicle.
Operations		
	detectFailure() : void	Checks if there was a failure to start task or a memory leak, and sets failure to False if so.
	restartRadio() : void	Restart the radio communication system.
	monitorThrottle() : void	Checks if there is a discrepancy between the command value to the throttle and the system context. Adjusts the throttle command if needed.
	communicateManeuver() : void	This function is called when the driver does not have time to safely stop before hitting an obstacle. The driver will be alerted to change lanes.
	trackTarget() : void	Command the camera to track the target vehicle in order to account for a vehicle going around a curve.
Relationships	VehicleController is a part of Vehicle. It also communicates with RadioCommunication, UserInterface, RadarSensor, ElectronicThrottle, GPS, Brake, Camera, and SecurityController.	

UML Extensions	VehicleController is a composition of Vehicle. RadioCommunication, RadarSensor, ElectronicThrottle, GPS, Brake, and Camera are aggregations of VehicleController. Vehicle can have zero or one Platoon and one VehicleController. SecurityController is a composition of VehicleController. VehicleController has an association with UserInterface. VehicleController can have one Vehicle, one RadioCommunication, one UserInterface, one to infinite RadarSensors, one ElectronicThrottle, one GPS, one Brake, one to infinite Cameras, and one SecurityController.

Sequence Diagrams:

Figure 4 represents scenario one. Figure 4 shows the scenario for when a vehicle is not in a platoon and chooses to enter the nearest platoon. If the vehicle is not in a platoon, it will request to join the nearest platoon. In turn, the platoon will add the vehicle to itself.

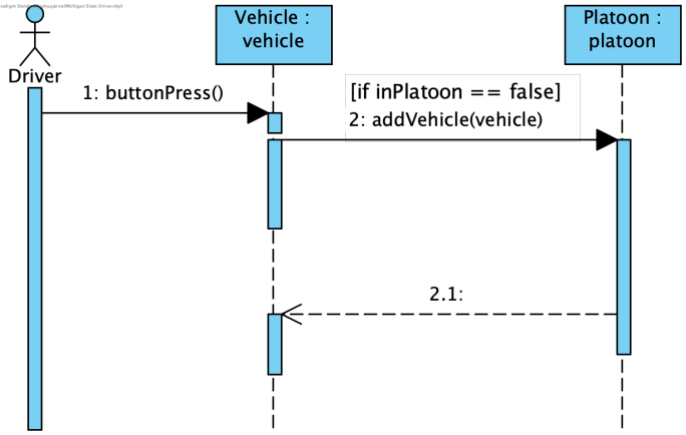

Figure 4: Sequence diagram for entering a platoon.

Figure 5 shows the scenario for when a vehicle is in a platoon and the driver chooses deactivate the CACC system. If the vehicle is in a platoon and the user presses the button to deactivate CACC, the vehicle will request to leave the platoon. In turn, the platoon will remove the vehicle from itself.

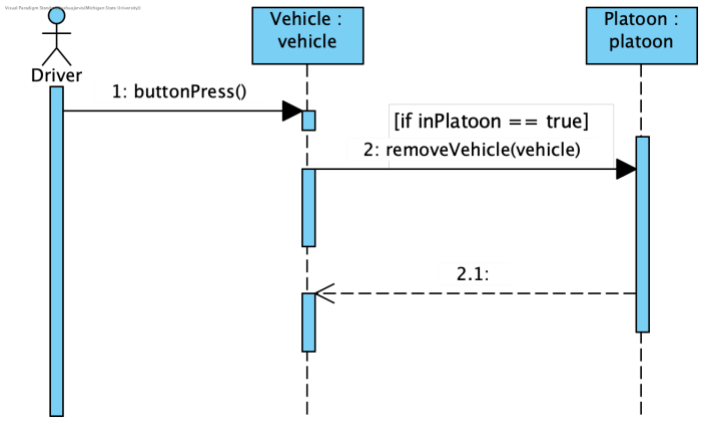

Figure 5: Sequence diagram for leaving a platoon.

Figure 6 shows the scenario for how the vehicle functions if it is not a part of a platoon the same scenario applies if the vehicle is the leader of a platoon. The vehicle controller will command the camera to detect for an object or target vehicle to follow. The vehicle controller will then command the radar sensors to detect the distance from the target vehicle. It will then calculate its safe stopping distance based on its current speed. The vehicle controller will command the brakes by wire to decelerate the vehicle while the safe stopping distance is less than the distance from the target vehicle. The vehicle controller will command the electronic throttle to accelerate the vehicle until the safe stopping distance is greater than the distance from the target vehicle.

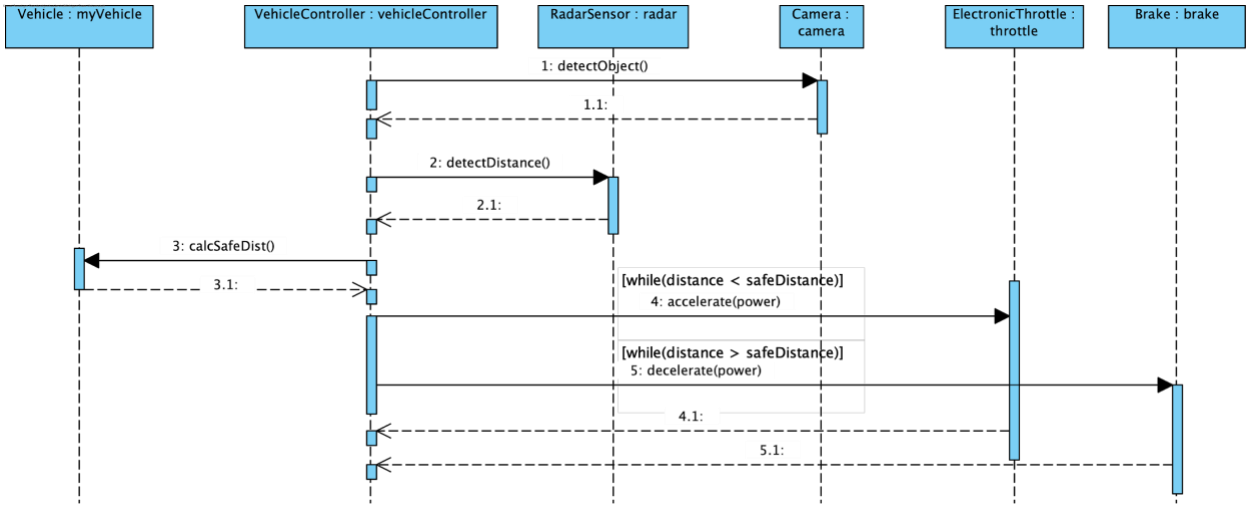

Figure 6: Sequence diagram for a vehicle that is not part of a platoon or is the leader of a platoon.

Figure 7 shows the scenario for when a vehicle is a follower in a platoon. The vehicle controller will command the camera to detect for a target vehicle to ensure that the target vehicle is there. The vehicle controller will command the radio communication to get the GPS object of the target vehicle. The vehicle controller will then find the target vehicle's location and direction using its GPS object and calculate the distance between itself and that vehicle. It will then calculate its safe stopping distance based on its current speed. The vehicle controller will command the

brakes by wire to decelerate the vehicle while the safe stopping distance is less than the distance from the target vehicle. The vehicle controller will command the electronic throttle to accelerate the vehicle until the safe stopping distance is greater than the distance from the target vehicle.

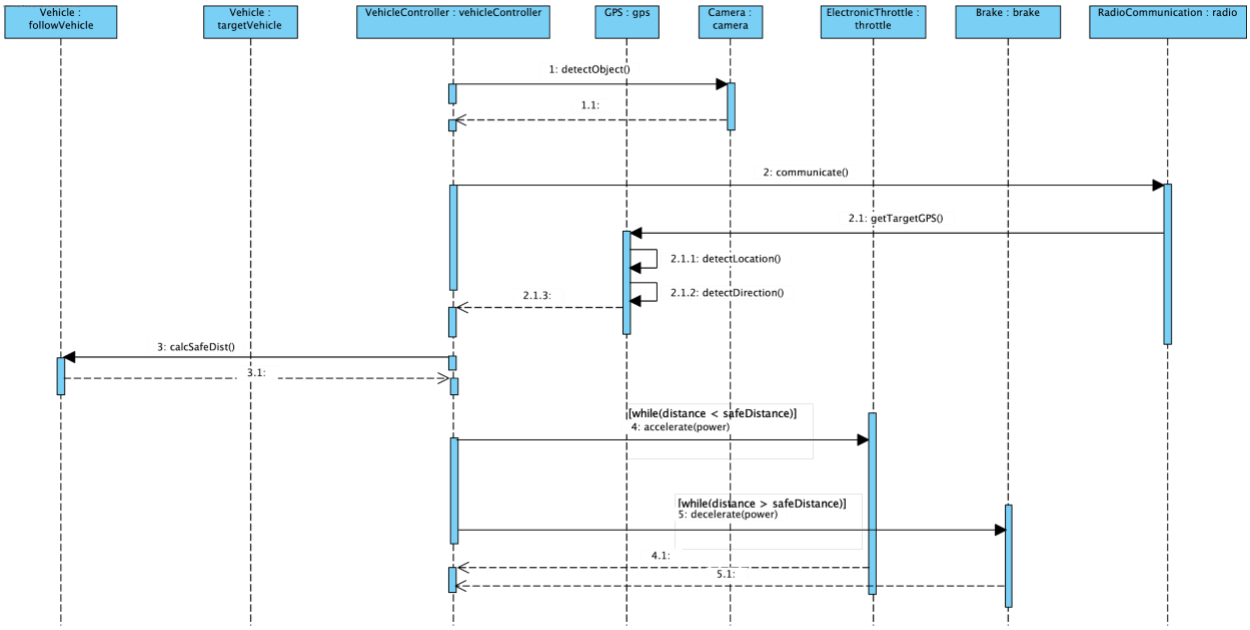

Figure 7: Sequence diagram for a vehicle that a follower of a platoon.

Figure 8 shows the scenario for when there is an object that is not a vehicle in front of the car. The vehicle controller will command the camera to detect for an object and the object's speed. If the object's speed is 0 mph, the vehicle controller will command the user interface to alert the driver that there is an obstacle that needs to be avoided. The vehicle controller will also command the brakes by wire to begin braking in order to offset the chance of hitting the obstacle.

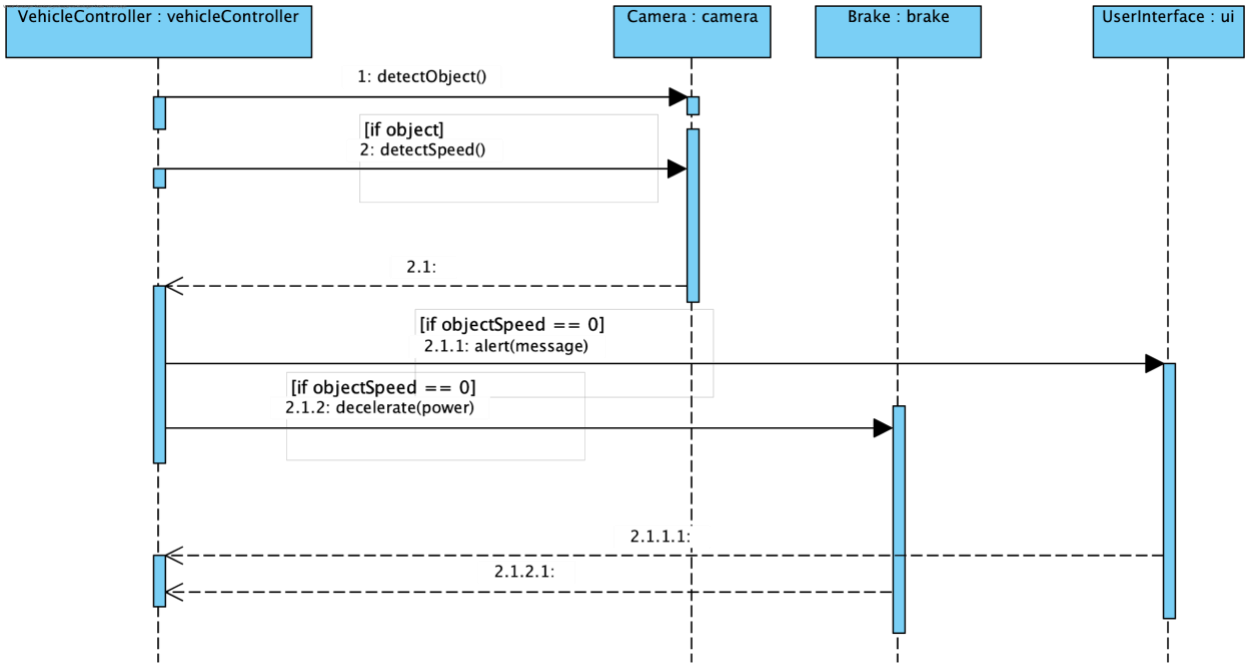

Figure 8: Sequence diagram for a vehicle that detects an obstacle ahead.

Figure 9 shows the scenario for a normal functioning vehicle in a two car platoon, and how the vehicle checks for failure to start task or a memory leak. Everytime the vehicle controller commands an object of the vehicle, it detects for a failure to start task and memory leaks. If there is no failure, the vehicle continues functioning with the CACC system. If there is a failure, however, the system will begin shutting down and a warning will be displayed to the user that there is a system failure.

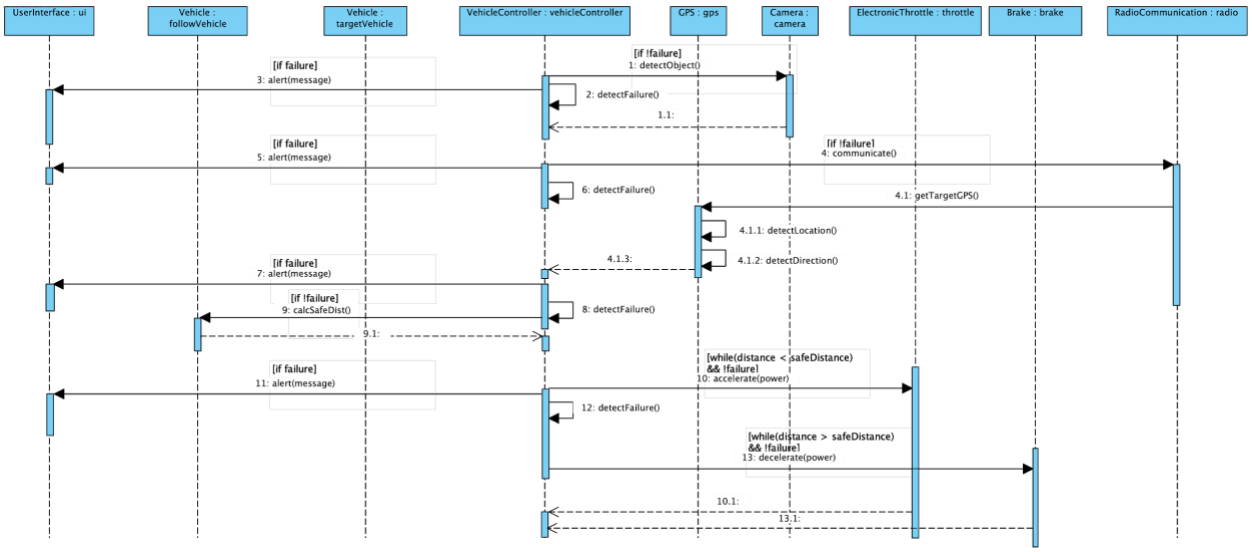

Figure 9: Sequence diagram for a vehicle in a platoon where the CACC system continuously checks for software failures.

Figure 10 shows the scenario for a normal functioning vehicle in a two car platoon, and the radio communication system fails. The vehicle controller will know if the radio communication system fails because when it tries to use the communication to obtain the GPS object of the target vehicle, no object will be returned. The vehicle controller then restarts the radio communication system.

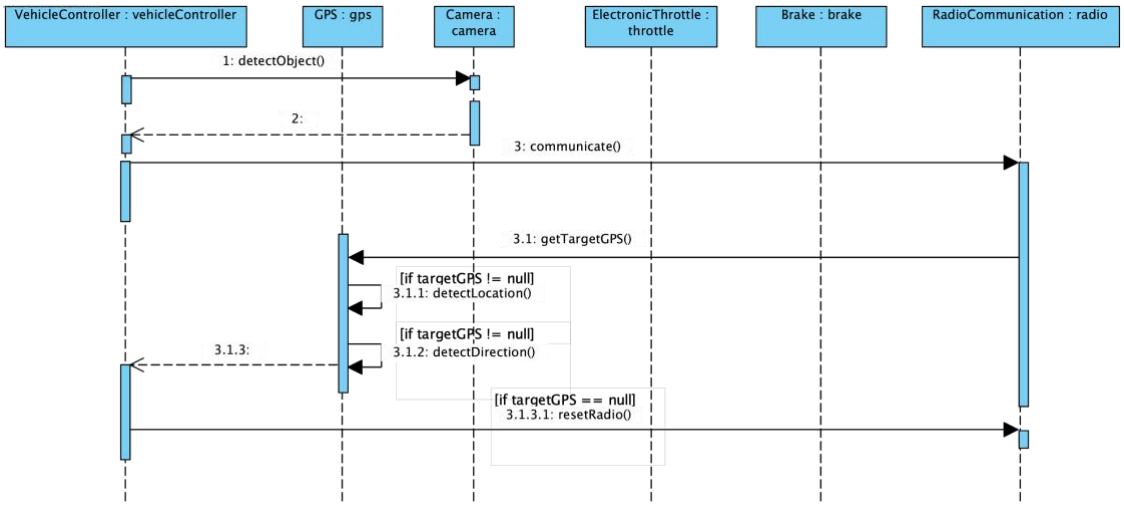

Figure 10: Sequence diagram for a vehicle in a platoon where the radio communication system fails.

Figure 11 shows the scenario for when the vehicle controller attempts to command the throttle to do an extreme acceleration that is not appropriate for the vehicle's given context. The vehicle controller monitors the throttle value and adjusts it if necessary before actually commanding the electronic throttle to accelerate.

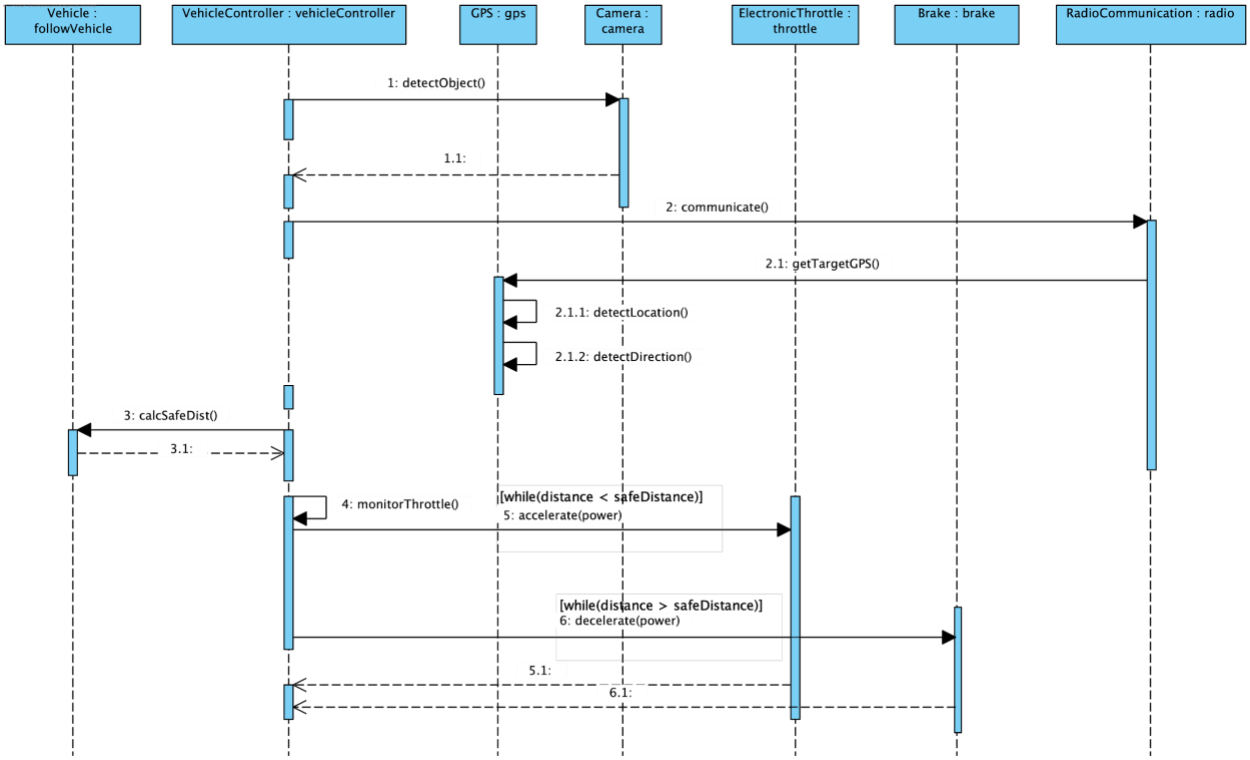

Figure 11: Sequence diagram for a vehicle in a platoon where the vehicle controller monitors the throttle values before commanding the electronic throttle.

Figure 12 shows the scenario where the leader of the platoon runs into a scenario where it must stop and then choose whether to tell the rest of the platoon to stop, or to move to adjacent open lanes. The vehicle controller will command the camera to detect an object. If the object has a speed of 0 mph, implying that it is an obstacle, the vehicle controller will command the radar sensors to detect the object's distance from the vehicle. The vehicle controller will command the user interface to alert the driver of an obstacle. The vehicle controller will then calculate the safe distance needed for stopping. If the safe distance is less than the distance from the obstacle, the vehicle controller will command the brakes by wire to decelerate the vehicle. If the safe distance is more than the distance from the obstacle, the vehicle controller will alert the driver to maneuver to a different lane and will command the radio system to communicate to the following vehicles to manually move to an adjacent lane.

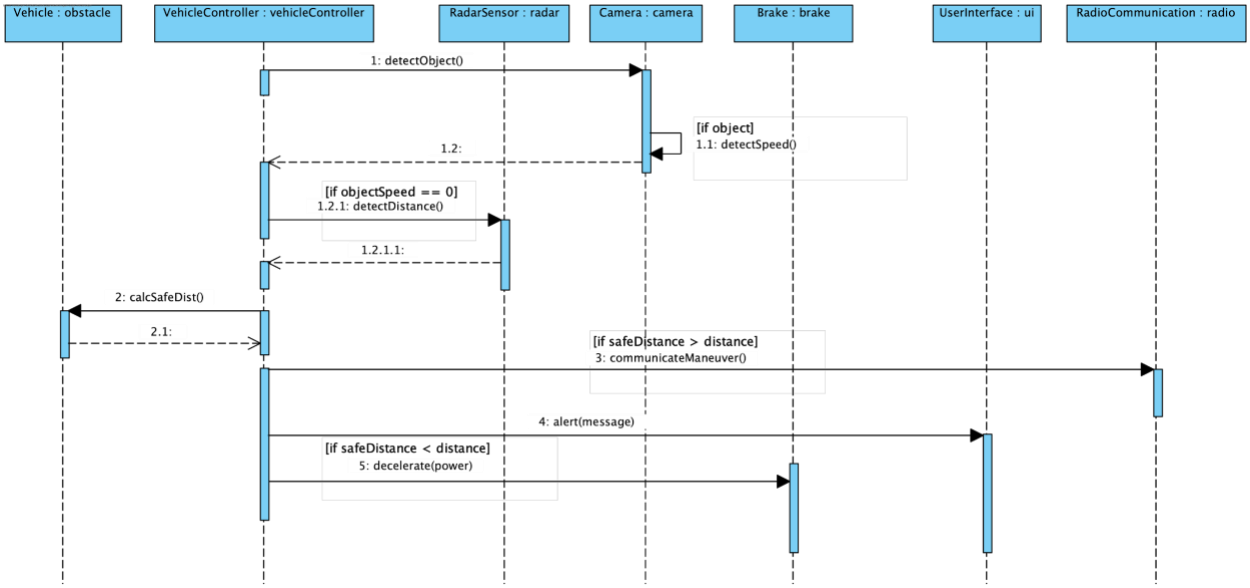

Figure 12: Sequence diagram for when a leading vehicle in a platoon needs to avoid an obstacle and it chooses whether to stop or move to an adjacent lane.

Figure 13 shows the scenario where the target vehicle is moving around a curve and the vehicles in the adjacent lane appear to be in front of the driving vehicle. The vehicle controller commands the camera to keep track of the target vehicle, so that it knows when the target vehicle is moving around a curve.

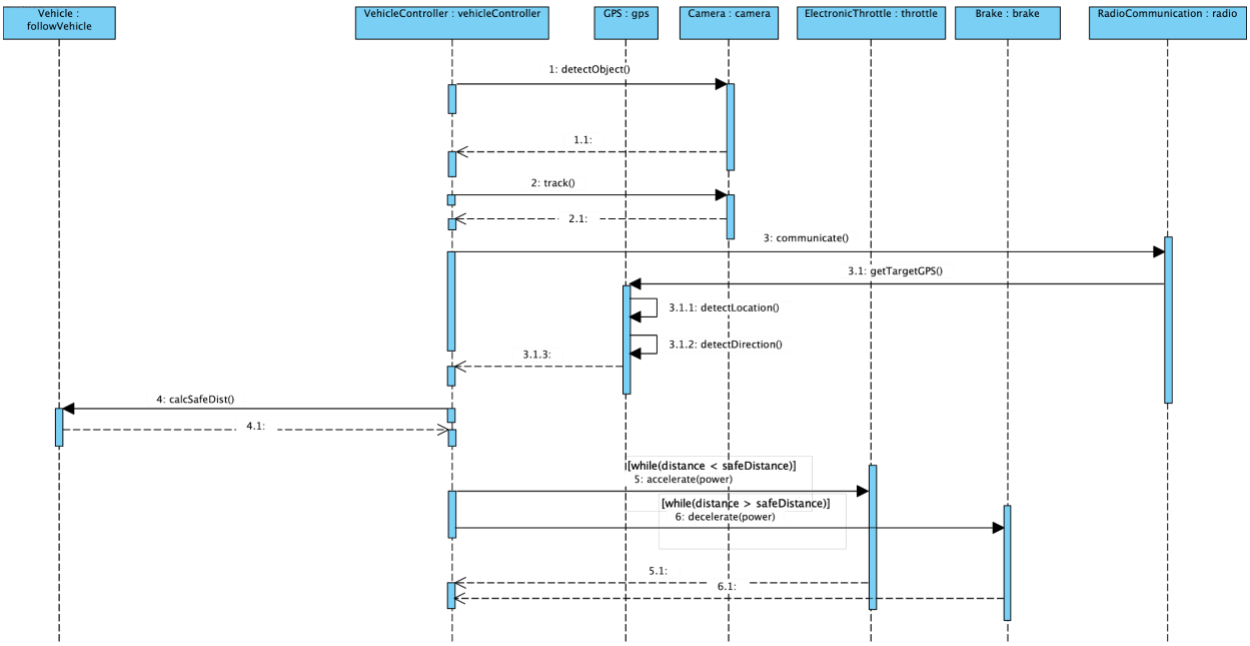

Figure 13: Sequence diagram for when the vehicle and its target vehicle drive around a curve.

Figure 14 shows the scenario for when the vehicles in a platoon are travelling up or down a hill and must maintain speed. The vehicle is continuously command the radar sensors to detect the distance behind the target vehicle. It will then calculate its safe stopping distance based on its current speed. The vehicle controller will command the brakes by wire to decelerate the vehicle while the safe stopping distance is less than the distance from the target vehicle. The vehicle controller will command the electronic throttle to accelerate the vehicle until the safe stopping distance is greater than the distance from the target vehicle.

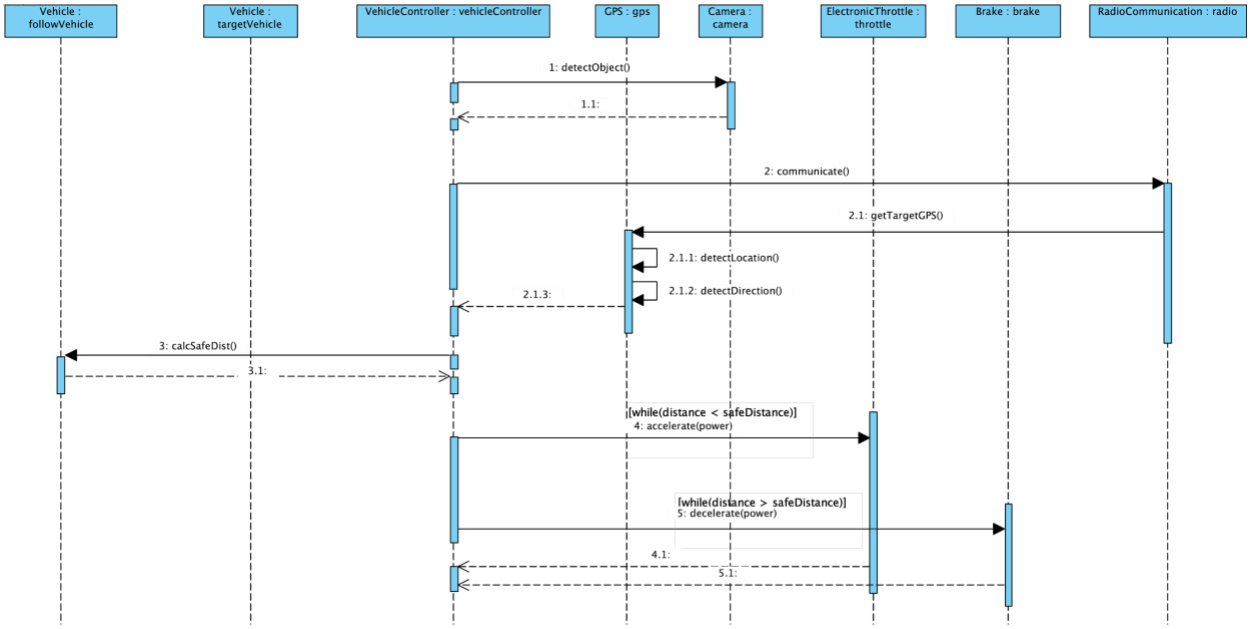

Figure 14: Sequence diagram for when the vehicle is travelling up or down a hill and must maintain its speed and distance from the target vehicle.

State Diagrams:

The camera system starts by searching for an object, if the object is found, it moves to the tracking object state. In the tracking object state, it senses the object's speed and moves into the sensing speed state. In the sensing speed state, if the object is lost, it switches back into the looking for object state.

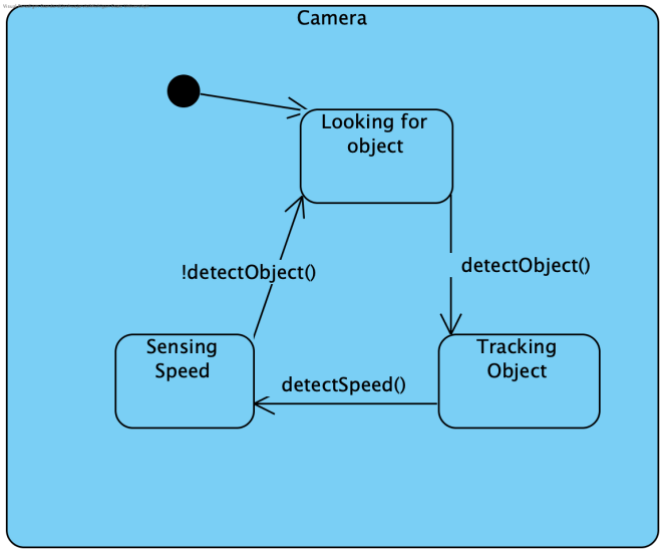

Figure 15: State diagram for camera.

The Electronic Throttle Control (ECT) can be in three main states: accelerating, maintaining speed, and not accelerating. It transitions between the states depending on the target and safe distances.

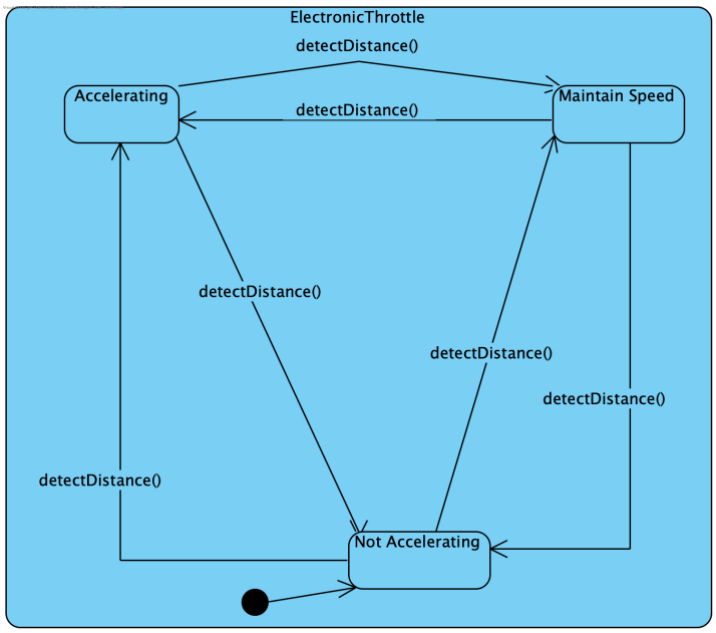

Figure 16: State diagram for ElectronicThrottle.

The GPS component transitions into two states simultaneously and remains active in those states. It continuously locates the vehicle and also calculates direction of travel.

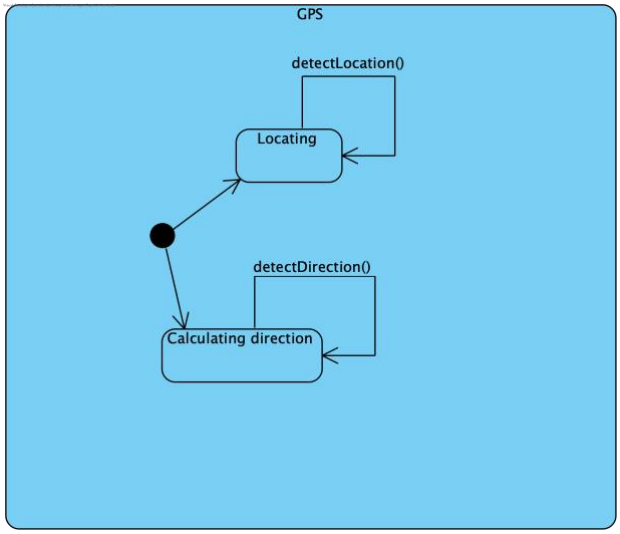

Figure 17: State diagram for GPS.

The Platoon's initial state is the created state when two vehicles join up. When the Platoon contains only the lead, the lead vehicle can disband the Platoon and therefore destroying it.

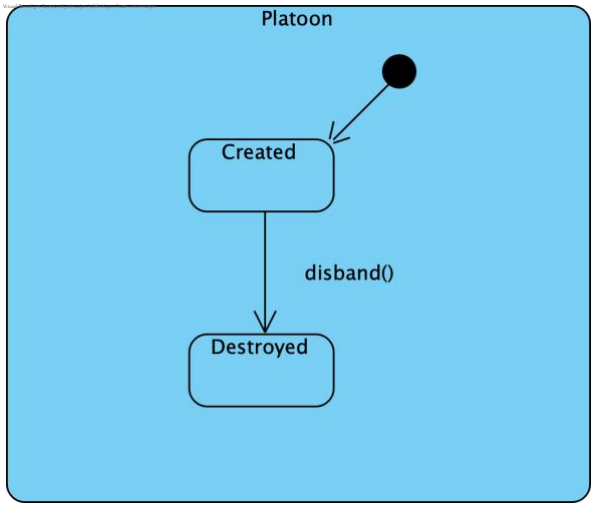

Figure 18: State diagram for Platoon.

The RadarSensor is in the sensing state when it is commanded by the VehicleController to sense the target vehicle. When the radarSensor is in the Sensing state, it continuously senses the target vehicle.

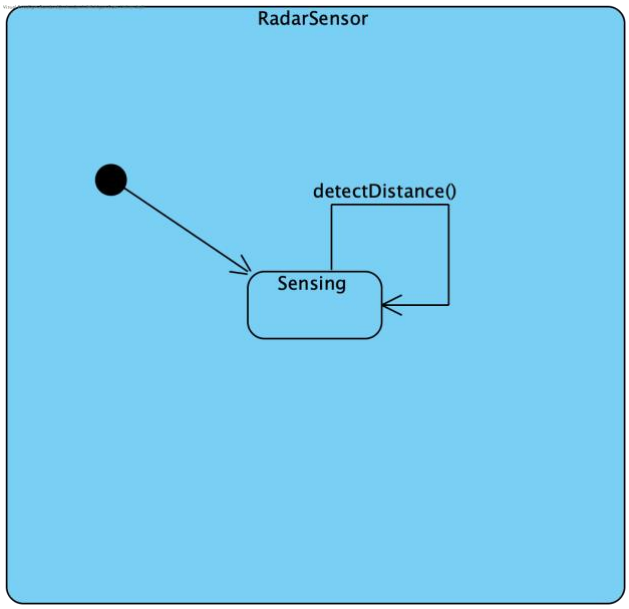

Figure 19: State diagram for RadarSensor.

The RadioCommunication system is initially in the receiving state, waiting to receive any information from a following vehicle. While in the receiving state, the radio system can be commanded by a following vehicle to transmit data to the following vehicle. The radio communication system is now in the transmitting state where it will transmit the vehicle data to another vehicle. While in the transmitting state, it can be commanded by the vehicle controller to

receive data from the target vehicle, transitioning the radio communication system to the receiving state.

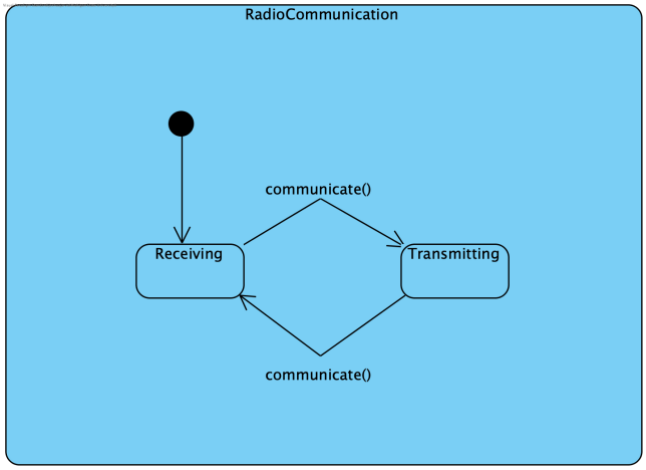

Figure 20: State diagram for RadioCommunication.

The security controller enters and constantly remains in the encrypting state while continuously encrypting the data.

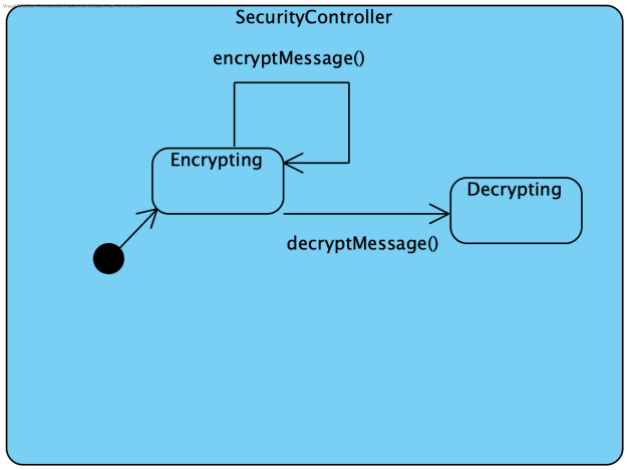

Figure 21: State diagram for SecurityController.

The UserInterface is initially in the No Alert state. In this state, no warning is displayed on the dashboard of the vehicle. If the user interface is commanded by the vehicle controller to alert the user it transitions to the Alerted state. In this state, the dashboard displays a warning message to the driver. After a timeout period, the user interface will remove the alert and return to the No Alert state.

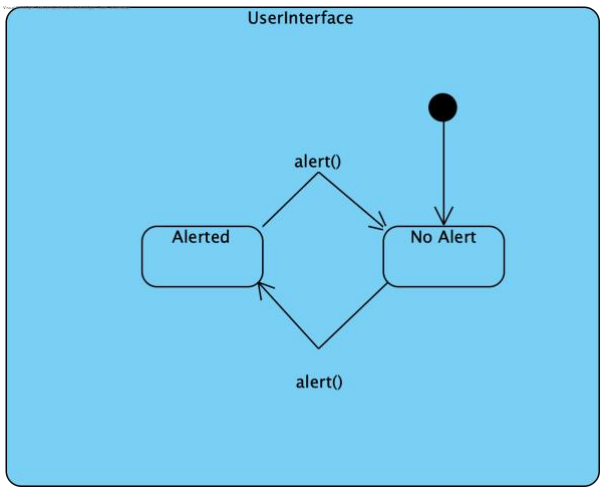

Figure 22: State diagram for UserInterface.

The vehicle starts with the CACC system not activated. If the system is turned on it is now activated and automatically in the solo state where it is not a part of a platoon. It can then search for and join a platoon to become in the platoon state. It can then leave the platoon to go back to the solo state. If the system is not activated, the user can turn on the system and it will be activated. The user can turn off the system and it will no longer be activated only if it's in the Solo or System On state.

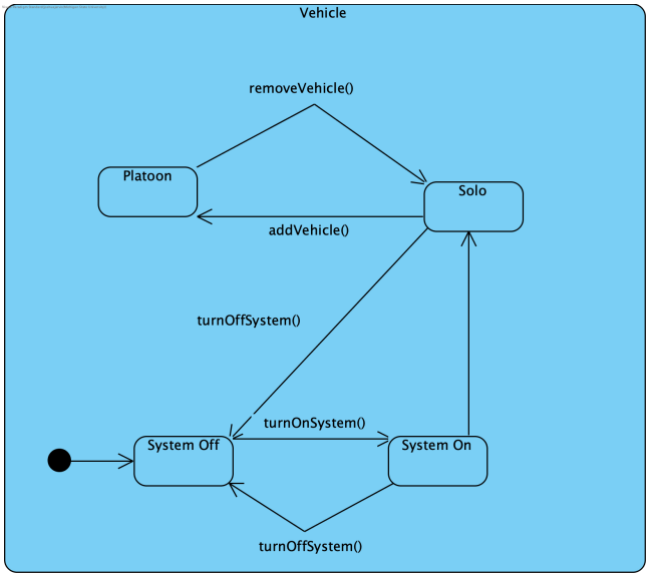

Figure 23: State diagram for Vehicle.

The vehicle controller simultaneously activates many sub-components and keeps those components constantly running. When the vehicle controller is activated, it activates the radio, user interface, radar, throttle, gps, brake, camera, and security controller systems. Then each sub-system constantly runs its routine.

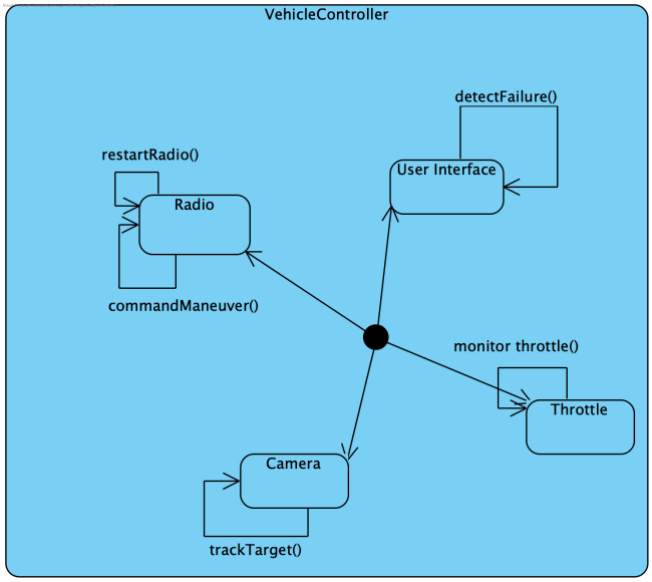

Figure 24: State diagram for VehicleController.

The brakes are initially not braking as the car is moving. If the distance between the target vehicle becomes greater than the safe distance, the brakes take the above safe distance transition to the braking state, where it begins braking. If the distance between the target vehicle becomes less than the safe distance, the brakes begin braking.

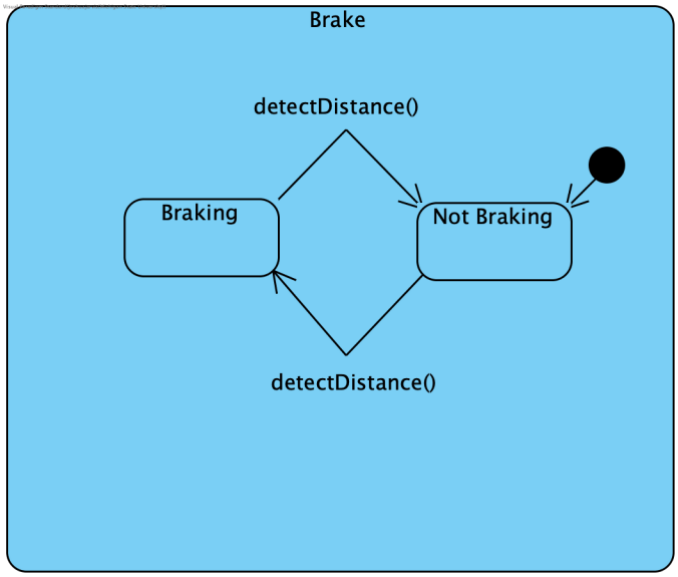

Figure 25: State diagram for Brake.

5. Prototype

Our prototype for the CACC system will display its main components and applications in a visually pleasing way. Proper documentation of the system includes a set of scenarios whereby the behavior of the system can be displayed. Our CACC prototype will describe some of these scenarios in detail, and provide an animation to accompany the description. The user can select a specific scenario with a simple press of a button, and both the description and animation will appear. Designing our prototype in this way will allow the user to better imagine how CACC works in the real world.

5.1. How to Run Prototype

Due to its location on our project website, prototype interaction will require access to the Internet. Our animations will be embedded Vimeo clips, which can only be viewed in browsers that can decode H.264 videos in an HTML5 player. Such browsers include: Chrome 30+, Firefox 27+, Internet Explorer 11, Microsoft Edge, and Safari 9+. Full functionality can only be guaranteed for operating systems that are still being supported by their proprietors. As of now, the animation begins immediately upon pressing a specific scenario button.

Link to Prototype v1: https://cse.msu.edu/~rajend16/prototypes/prototype1.html

5.2. Sample Scenarios

Ford Motor Company has provided us with a set of six scenarios to consider when outlining the behavior of CACC. One such scenario involves a vehicle's behavior in a platoon: if the lead vehicle in the platoon determines an emergency stop is necessary due to road conditions, how should the rest of the vehicles in the platoon behave (Scenario Four)? Can the platoon successfully stop or must the vehicles be diverted to adjacent lanes or the shoulder? The future version of our prototype will include an animation and description of this scenario amongst the rest, but for now, we have outlined the basic scenario of a vehicle entering a platoon.

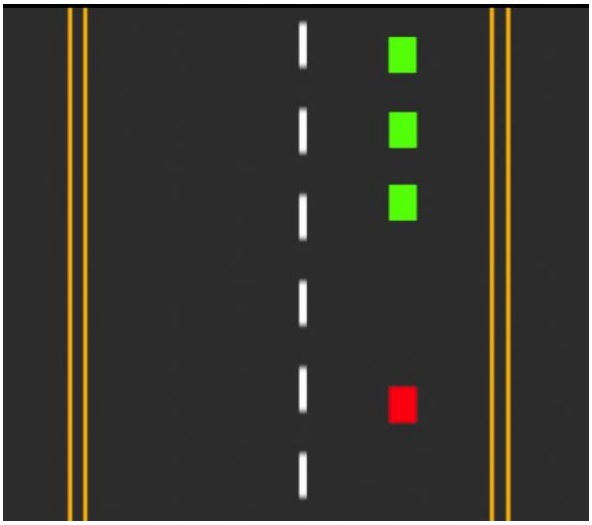

Figure 26: A vehicle (in red) just before entering a platoon (vehicles in green).

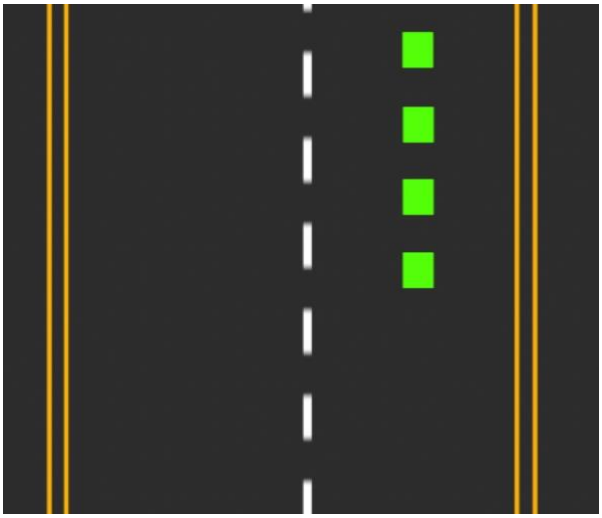

Figure 27: Vehicle after entering platoon.

6. References

[1] D. Thakore and S. Biswas, “Routing with Persistent Link Modeling in Intermittently Connected Wireless Networks,” Proceedings of IEEE Military Communication, Atlantic City, October 2005.

[2] Anayor, Chikamso, et al. “Cooperative Adaptive Cruise Control of A Mixture of Human Driven and Autonomous Vehicles.” SoutheastCon 2018, 2018, doi:10.1109/secon.2018.8479268.

[3] Chang, Ben-Jye, et al. “Cooperative Adaptive Driving for Platooning Autonomous Self Driving Based on Edge Computing.” International Journal of Applied Mathematics and Computer Science, vol. 29, no. 2, 2019, pp. 213–225., doi:10.2478/amcs-2019-0016.

[4] “Cooperative Adaptive Cruise Control.” Cooperative Adaptive Cruise Control | California Partners for Advanced Transportation Technology, path.berkeley.edu/research/connected-and-automated-vehicles/cooperative-adaptive-cruise-control.

[5] Ko, Wonshick, and Dong Eui Chang. “Cooperative Adaptive Cruise Control Using Turn Signal for Smooth and Safe Cut-In.” IEEE Computer, 13 Dec. 2018.

[6] Milam, William. “Cooperative Adaptive Cruise Control++ (CACC++).” Ford Motor Company.

[7] Milanes, Vicente, et al. “Cooperative Adaptive Cruise Control in Real Traffic Situations.” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 1, 2014, pp. 296–305., doi:10.1109/tits.2013.2278494.

[8] Shladover, Steven E., et al. “Cooperative Adaptive Cruise Control: Definitions and Operating Concepts.” Transportation Research Record: Journal of the Transportation Research Board, vol. 2489, no. 1, 1 Jan. 2015, pp. 145–152., doi:10.3141/2489-17.

[9] Shladover, Steven E et al. “Using Cooperative Adaptive Cruise Control (CACC) to Form High-Performance Vehicle Streams.” Berkeley California PATH Program, 2014, https://escholarship.org/uc/item/3m89p611.

[10] U.S. Department of Transportation. “Cooperative Adaptive Cruise Control Taking Cruise Control to the Next Level.” Federal Highway Administration, www.fhwa.dot.gov/publications/research/ear/16044/16044.pdf.

[11] Wang, Ziran, et al. “A Review on Cooperative Adaptive Cruise Control (CACC) Systems: Architectures, Controls, and Applications.” 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, doi:10.1109/itsc.2018.8569947.

[12] Rajendran, Manish. “Team CACC3’s Starter Site for CSE435.” 2019. http://cse.msu.edu/~rajend16/.

7. Point of Contact

For further information regarding this document and project, please contact **Prof. Betty H.C. Cheng** at Michigan State University (chengb at msu.edu). All materials in this document have been sanitized for proprietary data. The students and the instructor gratefully acknowledge the participation of our industrial collaborators.