Software Requirements Specification (SRS)

Project X

Authors:
Ethan Miller
Michael Dittman
Jason Kendall Harris
Jesse Barrett Stroster
Tianli Zhou

Customer: Andrew Davenport, GM

Instructor: Dr.Betty Cheng

1 Introduction

Driver fatigue is often colloquially cited as one of the major causes of automobile accidents, but it is hard to empirically study the magnitude of its impact on accident causation due to survivorship bias [1]. While driver fatigue is often thought of as the driver being drowsy on the road, the term actually encompasses various states of fatigue including what is known as Highway Hypnosis. Highway Hypnosis is when the environment the driver is operating in is monotonous and the layout of the road is relatively straight, leading the driver to enter a state where they have a higher response latency and decreased memory formation [2]. Any attempt to alleviate this phenomenon by automating driving in such an environment can potentially save many lives, which is why the development of a Hands Free Driving System (HFDS) is crucial to ensure highway safety.

This document will lay out the requirements for an HFDS. Each section will cover some of the purposes, requirements, descriptions, and diagrams within each system that is used throughout the process. The systems will be detailed in their relation to the overarching functionality of HFDS and illustrate the significance of each system and its ability to function together properly. The focus of the systems are specified and the importance of each system is highlighted in the models and diagrams. Each model and use case diagram will elaborate on the many potential situations given by the requirements list along with how each scenario will be handled. A prototype is also included to help visualize the system and its many functions, creating scenarios for the user to see how the system operates and how the many complicated systems come together.

1.1 Purpose

The purpose of this document is to describe the ins and outs of a Hands Free Driving System (HFDS). It will go into detail about the systems used, the features needed and the requirements to develop such a system. This documentation is created for the purpose of ensuring requirements are agreed upon by all related parties and is intended for stakeholders and developers of said system.

1.2 Scope

Hands Free Driving System (HFDS) is an embedded system within a vehicle that allows the driver to travel along a highway without having to use the steering wheel or the pedals. One of the many goals of HFDS is the ability for the driver to conveniently take their hands off the steering wheel and pedals, while still paying attention, and have the vehicle safely drive. Since the driver still has to pay attention, the system will ensure that the driver is not distracted while driving and can avoid the many hazards and dangers of driving, including Highway Hypnosis. The HFDS uses many subsystems to ensure not only the safety of the driver, but also safe road conditions for operation. Said subsystems include sensors like GPS, LiDAR, and RADAR to establish a safe path for the vehicle to follow along. When HFDS is enabled, eye and head movements are both monitored to confirm the driver's attentiveness even when their hands and feet aren't controlling the vehicle. Additionally, the car monitors road conditions, construction zones, position of lanes and other cars, and calculates if HFDS can be enabled. If HFDS cannot be enabled or must be turned off, multiple warnings will ensue to attempt to get the driver to regain control of the vehicle. If the multiple warnings are ignored and the driver is found to be not attentive, the car will turn the vehicle's hazard lights on, pull over, and stop the vehicle.

1.3 Definitions, acronyms, and abbreviations

- **Hands Free Driving System (HFDS) -** System to control a vehicle without the need of the driver touching the steering wheel
- **Global Positioning System (GPS) -** An accurate worldwide navigational and surveying facility based on the reception signals from an array of orbiting satellites
- **Light Detection and Ranging (LiDAR**) - A detection system which works on the principle of radar, but uses the light from a laser
- **Radio Detection and Ranging (RADAR) -** A system for detecting the presence of object by sending out pulses of high-frequency electromagnetic waves that are reflected off of the object back to the source
- **User Interface (UI) -** Input devices and software in which the user and computer subsystems interact
- **Human Machine Interface (HMI) -** The hardware or software through which an operator interacts with a controller.
- **Blue Path prediction system -** A specific path that a vehicle will travel along to not crash or switch lanes
- **Software Requirements Specification (SRS) -** A comprehensive description of the intended purpose and environment for software under development

1.4 Organization

The rest of the Software Requirements Specification (SRS) will highlight some of the key elements that make up the HFDS. Section 2 will feature an overall description of HFDS and will contain information about the functionality, constraints and assumptions. Section 3 will focus more on the requirements followed by section 4 which will dive deeper into the modeling and developmental aspects of the system. Section 5 will outline how to access the prototype, how to use it, and how it functions. Section 6 will contain citations to works referenced throughout the SRS and section 7 will provide the points of contact for developers and customers.

2 Overall Description

This section will give an overall description of the Hands-Free Driving System. It will cover such as product perspective, functionality, constraints, user characteristics, any assumptions and dependencies, and also an apportioning of requirements at the end. Diagrams will be included to help represent the constraints and functionality of the HFDS.

2.1 Product Perspective

Long road trips can be mentally taxing on a driver and while on the highway it can be tedious work as you cruise in one lane for hours to constantly keep up with the flow of traffic, tilt the steering wheel to follow the wind of the road, and tap the brakes on occasion when you run up too close behind the driver in front of you. With the Hands-Free Driving system it can allow a driver to relax a little more as the system will take control of the vehicle. Unlike normal cruise control where the vehicle will only keep a consistent speed, the HFDS will not only control the acceleration of the vehicle, but also will be able to steer and brake as the driver can be completely hands-off and an active passenger in their own car ready to take back the wheel whenever it is necessary.

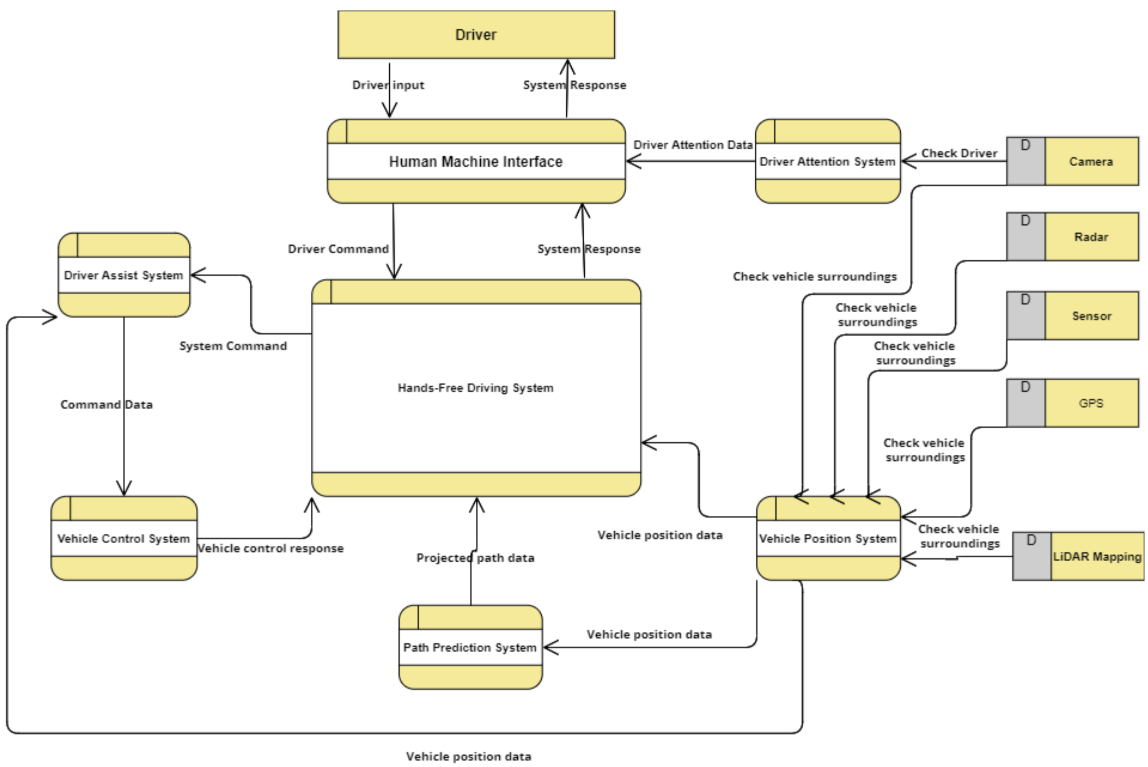

Figure 1.

In regards to software constraints for the Hands-Free Driving system, the system will be created off the adaptive cruise control that already exists in the vehicle. For the system to be activated adaptive cruise control must be enabled and the environment must be deemed safe by HFDS standards. The driver must set speed for adaptive cruise control, the HMI will take that driver input and allow for adjustments by driver whenever is necessary. Hardware constraints will include cameras inside and outside the vehicle. There will be a camera on the steering wheel to track the driver's eye and head movement to ensure the driver is actively paying attention to the road. Outside cameras will be used along with vehicle sensors, radar, and GPS to validate the vehicle's position in the real-world and the surroundings of the vehicle. With the position established the vehicle will use LiDAR mapping and information from the Vehicle Position system to continue driving in its current lane while being able to steer, brake, and accelerate automatically.

2.2 Product Functions

- "The HFDS system allows a vehicle to automatically steer, accelerate and brake in certain highway conditions." (Davenport)
- "The Driver Assist System validates road conditions, current trajectory, sensor input, and predicted path. If the current trajectory is deemed safe, the user can opt to enter hands-free mode." (Davenport)
- " The vehicle will enter an adaptive cruise control state and stay within its existing lane for the duration of the session." (Davenport)
- Camera monitoring will ensure that the driver is actively engaged on the road at all times
- Warnings will be sent to the driver if the driver is not properly engaged with the road. "If the system identifies the driver as inactive, the system aborts hands-free mode and if needed, the vehicle will come to a stop." (Davenport)

2.3 User Characteristics

Users are expected to own a driver's license and be in a good physical condition to operate a vehicle. The average user should have a comfortability with highway driving though the system works with any level of expertise on the road making the HFDS easy to use for all drivers. Users are expected to be actively engaged with the road while the system is active, in order to gain control at any time deemed necessary by the system or driver.

2.4 Constraints

The constraints for the HFDS system will ensure the safety of the driver and other drivers on the road. The HFDS cannot be activated unless the environment is deemed safe by the Driver Assist System and Adaptive Cruise Control is active. The driver must always have the ability to regain control of the vehicle by controlling the steering wheel, braking, or throttle. If the system detects any single point failure the system must relinquish control back to the driver. These are safety precautions to give the driver the ability to act if there is unforeseen obstacle the system has not detected. “Both hardware and sensor redundancies must be in place to ensure safe operation and provide time for the driver to become reengaged with the vehicle if a problem occurs.” (Davenport). For other safety precautions, if the lidar mapping does not match the projected path ahead, the system will relinquish control back to the driver. This will account for scenarios such as the vehicle driving into a construction zone where speed limits and lane closures can change on the fly. Security constraints will be in place to protect the system from software hacks while being able to accept over-the-air updates.

2.5 Assumptions and Dependencies

It is assumed that Lane detection and adaptive cruise control are pre-existing features within the vehicle. The HMI system will be able to work with adaptive cruise control and accept user input from the driver such as set speed and following distance. It can also be assumed that the vehicle is driving on the highway when activating the system. Another software assumption is LiDAR mapping of highways has been captured meaning paths are already in place for the vehicle to follow. The system should be able to work in moderate rain levels, any rain levels that obscure any sensors or cameras will be declared a fault and control will be relinquished back to the driver. In traffic situations the system will be able to lower the vehicle speed to match the flow of traffic, but will not raise the speed to exceed the speed limit. Users are expected to be actively engaged with the road at all times and ready to take back control of the vehicle at any point while the HFDS is active. The system is assumed to be able to run without the use of the internet. The Internet will only be a component when dealing with over-the-air updates.

2.6 Apprwoportioning of Requirements

In the future, the HFDS can include more features to ensure safety to users and to be able to adapt to new situations. The ability to detect emergency vehicles and safety constraints to handle over-the-air updates were discussed as points of interest to improve the current system’s functionality.

3 Specific Requirements

This section contains a list of requirements for the HFDS system. They are listed in order of importance, and are separated into global invariants, hardware, and security requirements.

1. **Global Invariant Requirement**
   1.1. The system must follow all speed limits
   1.2. If the systems detect any point of failure, control must be given back to the driver.
   1.3. The system must allow the driver to consistently regain control of the vehicle in multiple scenarios via controlling the steering wheel, braking, or throttle regardless of the state of the software
   1.4. The Driver Attention System must always be monitoring the driver’s head movement and eye movements to ensure active engagement.

Cameras must be able to monitor eye movements in all lighting conditions

1.5. The Vehicle Control System must always accept only legal actions to perform HFD.

1.6. The road which the user activates the HFDS must always have proper LiDAR mapping with the Path Prediction System.

1.7. The Vehicle Position System must always have the correct vehicle position, radar data, and camera data to detect, predict, and react to objects and cars in vicinity of the vehicle

1.8. Final warnings sent to driver that will send vibrations to driver to re-engage

1.9. If driver is deemed inactive, system must abort hands-free mode, and if needed come to a complete stop

1.10. Hardware and sensor redundancies must be in place to provide time for driver to become reengaged with the vehicle if a problem occurs

1.11. Steer, accelerate, brake automatically

1.12. System can perform all functions with a certain amount of non-functional sensors

1.13. Alert driver and relinquish control of the car in cases when an error or fault cannot be corrected by the equipment

1.14. If there is any fault whatsoever i.e sensor turns off, sensor is damaged, any subsystem is not active, and driver is inattentive after 3 warnings, car should turn on hazard lights and come to a complete stop

2. Primary Requirements

2.1. Hardware

- 2.1.1. Long range radars to detect following distance of other pedestrians, objects, and vehicles when HFDS is enabled
- 2.1.2. Separate backup sensors to account for any redundancy when a primary sensor fails
- 2.1.3. Gps sensors to navigate the vehicle to the desired location
- 2.1.4. Cameras to detect when driver may be distracted
- 2.1.5. Human interface display to turn on and off the HFDS system, display warnings and the vehicles current trajectory
- 2.1.6. Internet connection hardware to ensure system receives over the air updates

2.2. Software

- 2.2.1. Objects in front and around the vehicle using radars, gps, and sensors to determine if they are moving or static
- 2.2.2. Camera images of driver should be reliably processed to determine if they are distracted
- 2.2.3. The system should be efficient and easy to use
- 2.2.4. The system should meet all industry programming standards

2.3. Cyber Security Requirements

- 2.3.1. The system should be able to receive over the air updates published only by the manufacturer or vendor

2.3.2. Any and all connections to a network must remain encrypted
2.3.3. The system must be regularly tested to prevent most DOS or other compromising attacks
2.3.4. The system should be able to detect when it has been compromised and immediately hand control to the driver

4 Modeling Requirements

Section 4 depicts several types of diagrams which model the requirements of the HFD system that are specified in Section 3. Illustrated in the following subsections are the key components, functionality and scenarios associated with the HFD system.

4.1 Use Case Diagram

Section 4.1 outlines the use case diagram for the HFD system. Use case diagrams are used to depict the behavior and operation of a system. They describe what happens when the system is running. Use case diagrams display one feature of a system or one set of interactions between the system and the user. They can depict boundaries, actors, functionality and outcomes of actions.

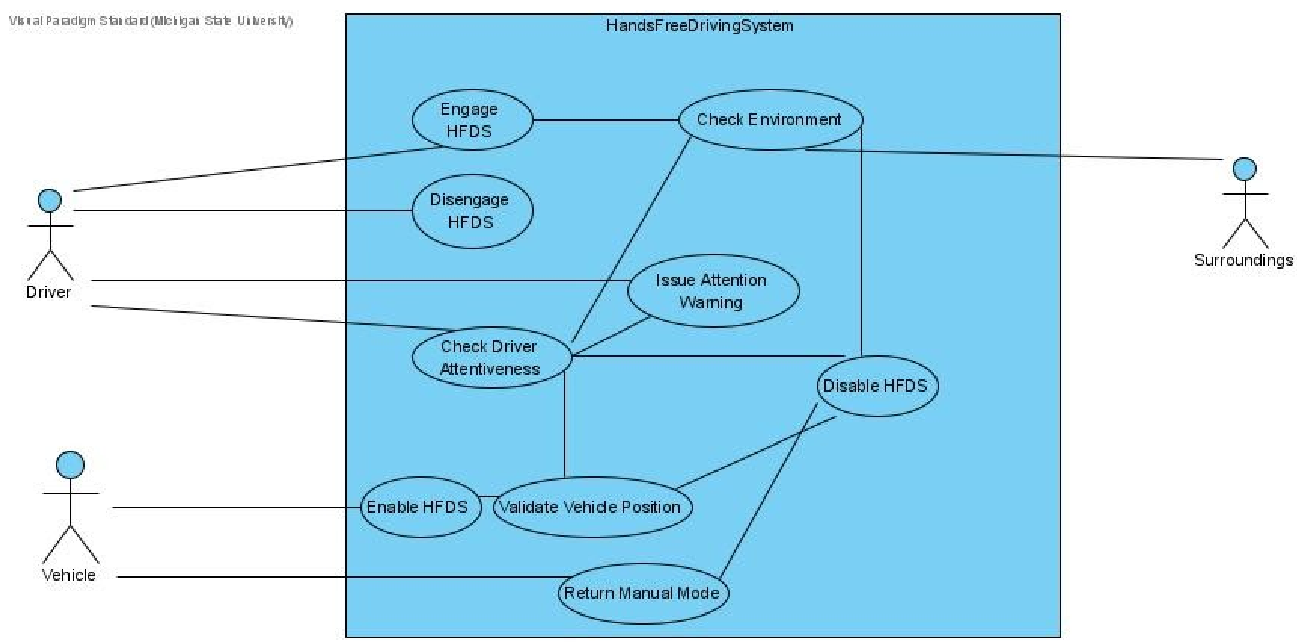

Figure 2. Use Case Diagram for HFDS

4.2 Domain Model and Data Dictionary

Section 4.2 outlines the domain model and data dictionary for the HFD system. The domain model is used to focus on the structure of a system as opposed to outcomes of actions. Domain models are used to highlight the key elements of a system.

4.2.1 Domain Model

The domain model in Figure x outlines the structure of the HFD system. Each of the blue boxes in Figure x represents a class in the system. Each class in the diagram is an object that interacts and has a role to play in the system. Relationships between classes are drawn as a line between classes. The relationships indicate that the two objects interact with each other in the system. The number of objects that exist in a given association is indicated by the number written near the endpoints of the relationship line. For instance there can only be one of every system and subsystem, but there can be any number of warning outputs and actuators (any number indicated by the asterisk). The relationships between objects can be further specified by aggregation and composition. Aggregation shows that an object is part of another object. This is shown as an open diamond at the endpoint of the object that has other objects as part of itself. As an example, four subsystems are part of the Hands Free Driving System. Composition shows that an object is owned by another object. This is shown as a shaded diamond terminated at the owning class. Composition is not featured in this domain model. Further, classes can exhibit generalization which depicts the relationship between two objects when one object, the child, can be said to be the same type of object as the parent object. Classes can have any number of attributes and operations. Attributes are named values of classes and operations are services that can be requested of the class. The HFD system is composed of six main subsystems: the Human Machine Interface, the Driver Attention System, the Driver Assist System, the Vehicle Position Subsystem, the Path Prediction System, and the Vehicle Control System as shown in Figure X. The HFD system itself only has relations to four of the subsystems. The diagram further features classes for actuators, position sensors, and warning outputs.

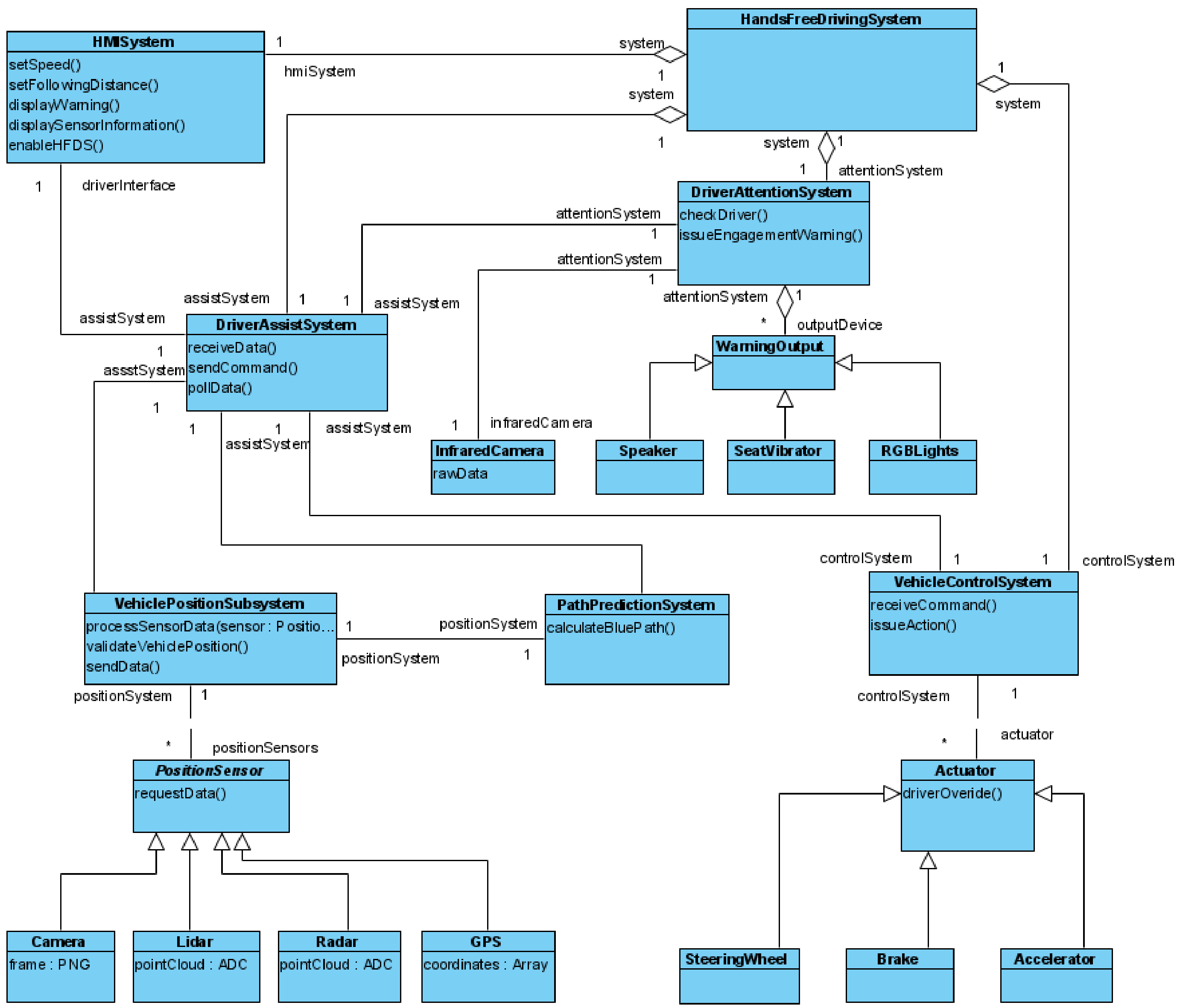

Figure 3. High-Level Class diagram (Domain Model) for HFDS

4.2.2 Data Dictionary

The following tables elaborate each of the classes that exist in the above domain diagram. Classes will be elaborated to show their relationships, operations, and attributes. The dictionaries will depict: aggregation, composition, attributes, operations, and generalization. Aggregation shows that an object is part of another object. This is shown as an open diamond at the endpoint of the object that has other objects as part of itself. As an example, four subsystems are part of the Hands Free Driving System. Composition shows that an object is owned by another object. This is shown as a shaded diamond terminated at the owning class. Composition is not featured in this domain model. Further, classes can exhibit generalization which depicts the relationship between two objects when one object, the child, can be said to be the same type of object as the parent object. Classes can have any number of attributes and operations. Attributes are named values of classes and operations are services that can be requested of the class.

Element Name		**Description**
Accelerator		An actuator that is physically responsible for the acceleration of the vehicle.
Attributes	None	
Operations	None	
Relationships	Associations : None Aggregations : None Composition : None Generalization: Is a type of actuator	
UML Extensions	None	

Table 1. Data dictionary entry for *Accelerator*

Element Name		Description
Actuator		A device that the user can interact with to change the state of the Hands Free Driving System.
Attributes	None	
Operations	driverOveride()	All actuators can be interacted with to disengage the Hands Free Driving System.
Relationships	Associations : None Aggregations: Is part of the VehicleControlSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 2. Data dictionary entry for *Actuator*

Element Name		**Description**
Brake		An actuator that is physically responsible for the deceleration of the vehicle.
Attributes	None	
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of actuator	
UML Extensions	None	

Table 3. Data dictionary entry for *Brake*

Element Name		Description
Camera		Position sensor responsible for recording visual data about the surrounding environment.
Attributes	frame : PNG	A single frame of visual data stored as a png file.
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of PositionSensor	
UML Extensions	None	

Table 4. Data dictionary entry for *Camera*

Element Name		**Description**
DriverAssistSystem		System that polls necessary information to determine safe conditions. Responsible for issuing commands to DriverControlSystem.
Attributes	None	
Operations	receiveData()	Service that receives data from VehiclePositionSubsystem and PathPredictionSystem.
	pollData()	Service that requests data from the VehiclePositionSubsystem.
	sendCommand()	Service that sends commands to the VehicleControlSystem.
Relationships	Associations : HMISystem, DriverAttentionSystem, VehicleControlSystem, PathPredictionSystem, VehiclePositionSubsystem Aggregations: Is part of HandsFreeDrivingSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 5. Data dictionary entry for *DriverAssistSystem*

Element Name		**Description**
DriverAttentionSystem		Subsystem that monitors the driver's engagement with the road.
Attributes	None	
Operations	checkDriver()	Service that checks the driver's engagement with the road.
	issueEngagementWarning()	Service that tells the warning outputs to activate.
Relationships	Associations : DriverAssistSystem, InfraredCamera Aggregations: Is part of HandsFreeDrivingSystem, Is composed of WarningOutput Compositions : None Generalization: None	
UML Extensions	None	

Table 6. Data dictionary entry for *DriverAttentionSystem*

Element Name		**Description**
GPS		System that monitors the position via satellites.
Attributes	coordinates : Array	An array of the coordinates of the vehicle.
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a PositionSensor	
UML Extensions	None	

Table 7. Data dictionary entry for *GPS*

Element Name		**Description**
HMISystem		System that displays sensor information and issues various warnings when needed.
Attributes	None	
Operations	setSpeed()	Service that allows the user to set the speed.
	setFollowingDistance()	Service that allows the user to specify the following distance.
	displayWarning()	Service that displays warnings to the user.
	displaySensorInformation()	Service that displays sensor information to the user.
	enableHFDS()	Service that allows the user to enable HFDS.
Relationships	Associations : DriverAssistSystem Aggregations: Is part of HandsFreeDrivingSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 8. Data dictionary entry for *HMISystem*

Element Name		**Description**
HandsFreeDrivingSystem		The main system that keeps the vehicle in a highway lane automatically.
Attributes	None	
Operations	None	
Relationships	Associations : DriverAssistSystem Aggregations: Is made up of : HMISystem, DriverAssistSystem, DriverAttentionSystem, VehicleControlSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 9. Data dictionary entry for *HandsFreeDrivingSystem*

Element Name		Description
InfraredCamera		Device responsible for measuring the attentiveness of the driver.
Attributes	rawData	The raw analog data the camera collects
Operations	None	
Relationships	Associations : DriverAttentionSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 10. Data dictionary entry for *InfraredCamera*

Element Name		**Description**
Lidar		Position sensor responsible for collecting and providing light detection data.
Attributes	pointCloud : ADC	The raw data stored as a point cloud in an ADC format.
Operations	None	
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a PositionSensor	
UML Extensions	None	

Table 11. Data dictionary entry for *Lidar*

Element Name		Description
PathPredictionSystem		Subsystem responsible for calculating the projected path based on information from the VehiclePositionSubsystem.
Attributes	pointCloud : ADC	The raw data stored as a point cloud in an ADC format.
Operations	calculateBluePath()	Function that calculates the projected path based on data from the VehiclePositionSubsystem.
Relationships	Associations : VehiclePositionSubsystem, DriverAssistSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 12. Data dictionary entry for *PathPredictionSystem*

Element Name		Description
PositionSensor		Class that defines the functionality for all types of position sensors available to the HFDS.
Attributes	None	
Operations	requestData()	Function that requests data from the relevant vehicle sensor.
Relationships	Associations : VehiclePositionSubsystem, DriverAssistSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 13. Data dictionary entry for *PositionSensor*

Element Name		Description
RGBLights		Responsible for controlling the flashing warning lights for the HFDS.
Attributes	None	
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of WarningOutput	
UML Extensions	None	

Table 14. Data dictionary entry for *RGBLights*

Element Name		**Description**
Radar		Position sensor responsible for collecting and delivering radar data.
Attributes	pointCloud : ADC	The raw data stored as a point cloud in an ADC format.
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of Position Sensor	
UML Extensions	None	

Table 15. Data dictionary entry for *Radar*

Element Name		**Description**
SeatVibrator		Warning output responsible for vibrating the seat of the user.
Attributes	None	None
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of WarningOutput	
UML Extensions	None	

Table 16. Data dictionary entry for *SeatVibrator*

Element Name		Description
Speaker		Warning output responsible for playing warning sounds to the user.
Attributes	None	None
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of WarningOutput	
UML Extensions	None	

Table 17. Data dictionary entry for *Speaker*

Element Name		Description
SteeringWheel		The actuator responsible for the angle along the transverse axis of the vehicle.
Attributes	None	None
Operations	None	None
Relationships	Associations : None Aggregations: None Compositions : None Generalization: Is a type of Actuator	
UML Extensions	None	

Table 18. Data dictionary entry for *SteeringWheel*

Element Name		**Description**
VehicleControlSystem		The system responsible for issuing commands to the vehicle actuators.
Attributes	None	None
Operations	receiveCommand()	Function that receives a command from the DriverAssistSystem.
	issueAction()	Function that tells an actuator the action to take.
Relationships	Associations : DriverAssistSystem Aggregations: Is part of HandsFreeDrivingSystem Compositions : None Generalization: None	
UML Extensions	None	

Table 19. Data dictionary entry for *VehicleControlSystem*

Element Name		**Description**
VehiclePositionSubsystem		The system responsible for processing sensor data from the vehicle's position sensors.
Attributes	None	None
Operations	processSensorData(sensor: PositionSensor)	Function responsible for processing any sensor data from any of the position sensors.
	validateVehiclePosition()	Function responsible for validating the vehicles position with respect to the position sensors and calculated blue path.
	sendData()	Function responsible for sending data from the position sensors to the DriverAssistSystem.
Relationships	Associations : PathPredictionSystem, DriverAssistSystem Aggregations: None Compositions : None Generalization: None	
UML Extensions	None	

Table 20. Data dictionary entry for *VehiclePositionSubsystem*

<table>
<tr><th colspan="2">Element Name</th><th>Description</th></tr>
<tr><td colspan="2">WarningOutput</td><td>The device which defines all actions of all warning outputs.</td></tr>
<tr><td>Attributes</td><td>None</td><td></td></tr>
<tr><td>Operations</td><td>None</td><td></td></tr>
<tr><td>Relationships</td><td colspan="2">Associations : None
Aggregations: Part of DriverAttentionSystem
Compositions : None
Generalization: None</td></tr>
<tr><td>UML Extensions</td><td colspan="2">None</td></tr>
</table>

Table 21. Data dictionary entry for *WarningOutput*

4.3 Sequence Diagrams

This section will introduce sequence diagrams for specific scenarios in the HFD system. Sequence diagrams are used to capture the step by step procedure by which the system executes one of its features. Sequence diagrams show how objects interact with each other via the ordering of messages. The blue boxes along the top of the sequence diagram represent objects. Arrows between them indicate a message being sent between objects. The sequence diagram shows progression from top to bottom, messages towards the top of the diagram happen first and messages towards the bottom happen last.

4.3.1 Normal Operation

The following sequence diagram shows what happens when the system is fully operational with proper road conditions. It demonstrates the user activating the system when the vehicle is on a supported highway. Also demonstrated is how the system interacts with the DriverAttentionSystem. First, the driver turns on HFDS. Then the system checks if the Adaptive Cruise Control is on and ensures that the conditions are safe. If the driver is not paying attention, lights are displayed to ensure that they are paying attention. If the driver continues to not pay attention, another warning is displayed. After the third warning period, HFDS is turned off and the car pulls over.

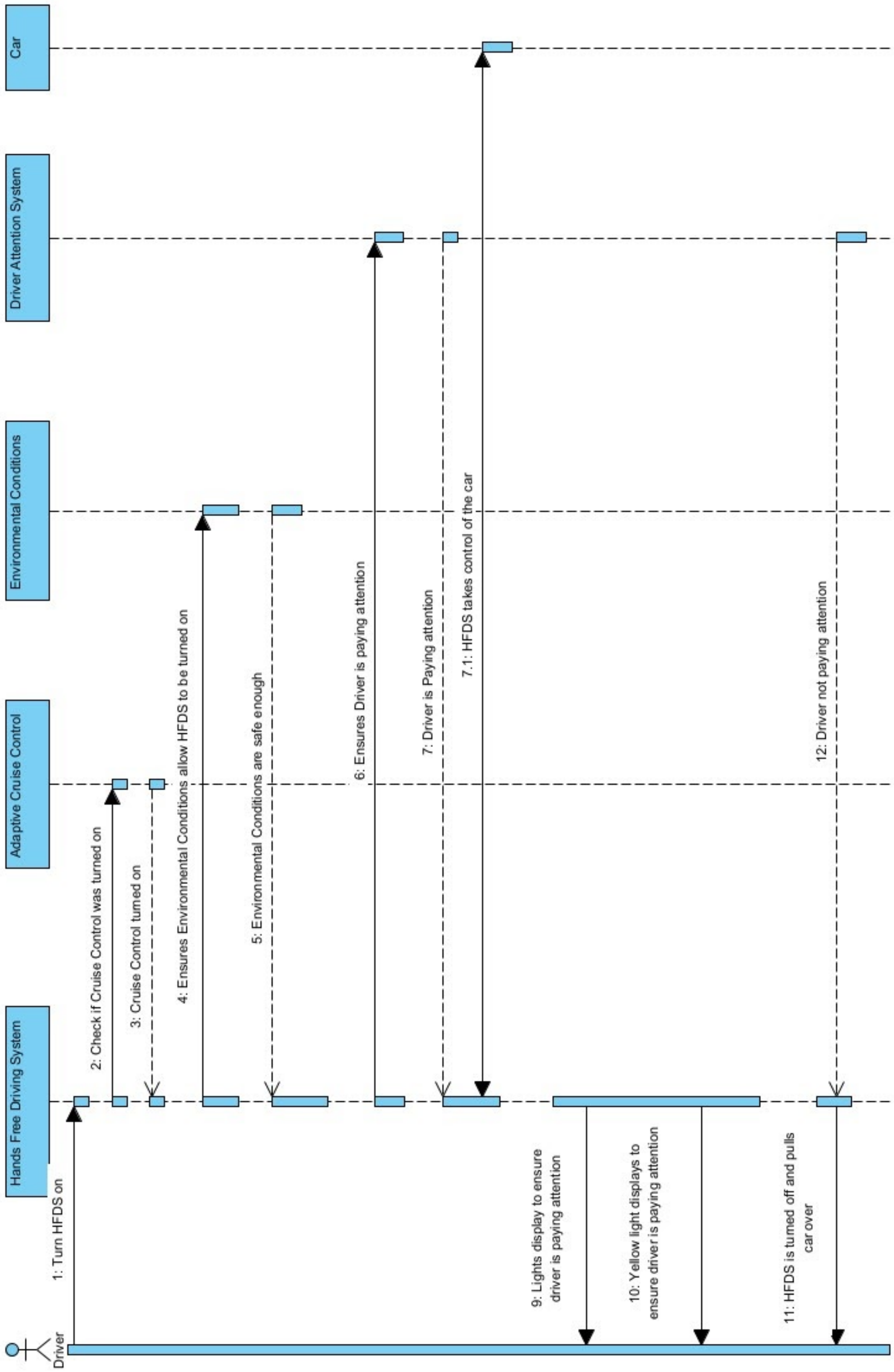

Figure 4. Sequence Diagram for Turning On the System and under normal operation

4.3.2 System Failure

The following sequence diagram shows what happens when there is a system failure. It demonstrates the user activating the system when the vehicle is on a supported highway. Also demonstrated are how the system interacts with the DriverAttentionSystem. First the driver turns on HFDS. Then the system checks if the Adaptive Cruise Control is on and ensures that the conditions are safe. The system then checks to see if there is a system failure. If there are no failures detected the system exhibits nominal behavior. When a failure occurs the system immediately warns the driver and HFDS deactives bringing the car to a complete stop with warning lights active.

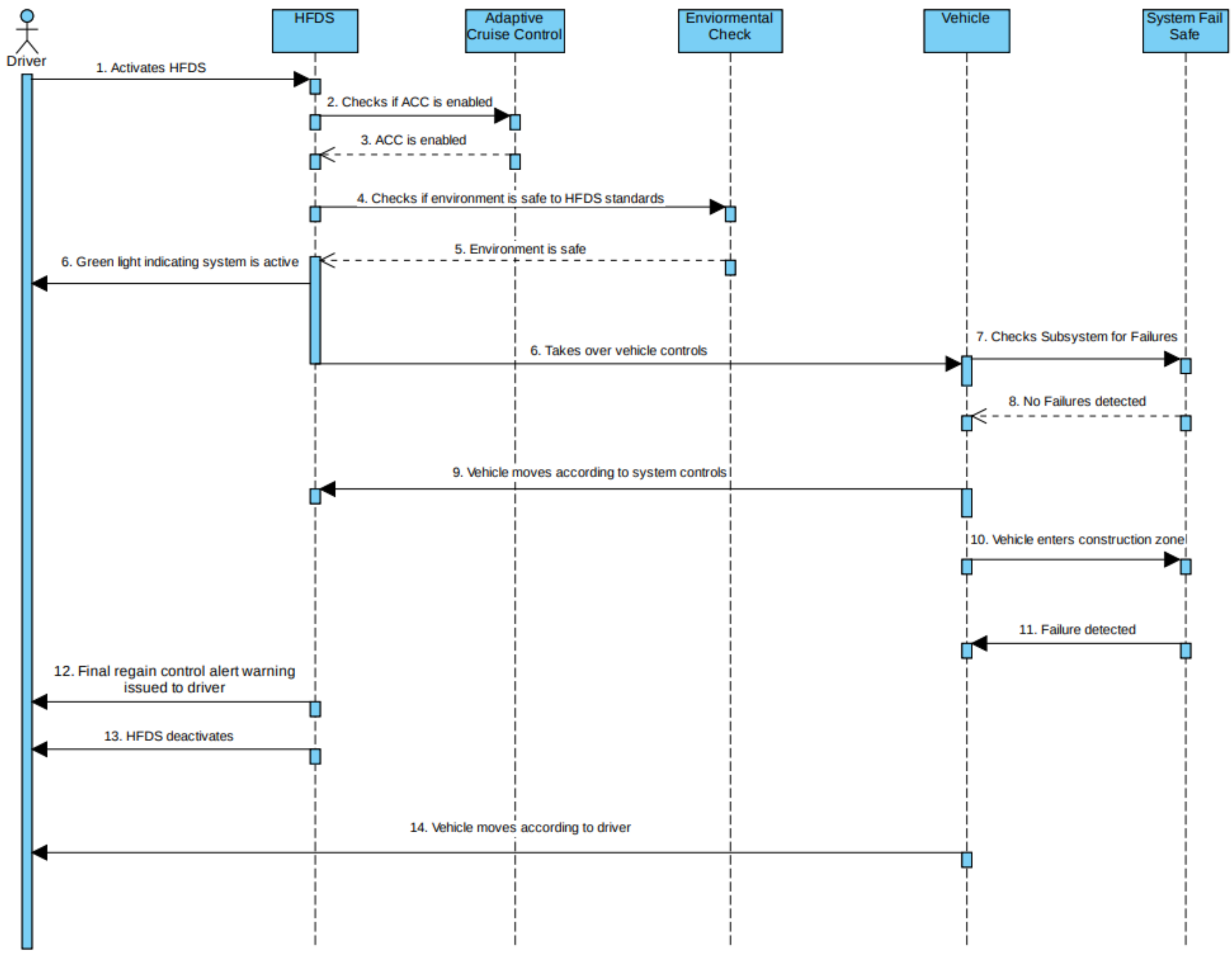

Figure 5. Sequence Diagram for A Single System Failure

4.3.3 Driver Disengagement

The following sequence diagram depicts what happens when the driver becomes permanently disengaged from the road. First if the driver is found to not be attentive for 5 seconds the first warning is issued. If the driver is found to not be attentive for another 5 seconds the third warning is issued. After another 5 seconds the third and final warning is issued, vibrating the seat of the user. After another 5 seconds if the driver is still not attentive the car will come to a complete stop.

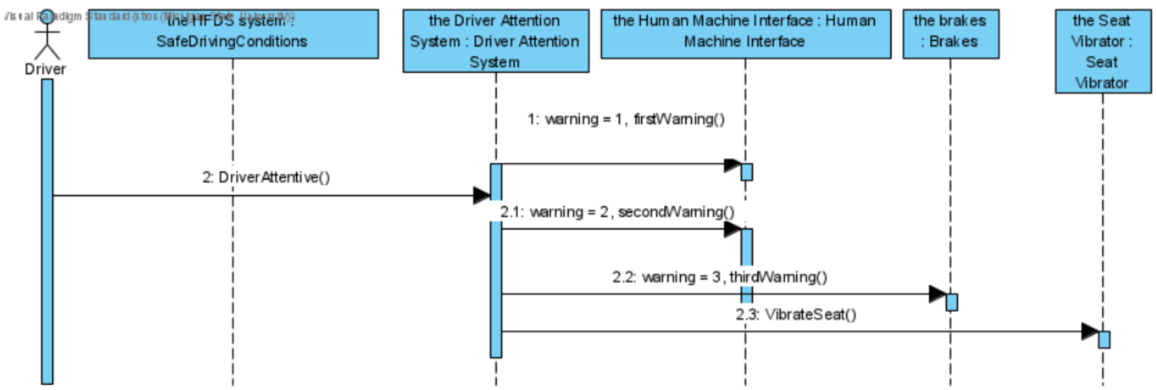

Figure 6. Sequence Diagram for Permanent User Disengagement

4.3.4 Continuous Control Change

The following sequence diagram shows what happens when there is a continuous change of control back and forth between the driver and HFDS. First the driver enables HFDS which tells the HFDS system to turn on. If enableHFDS() is false, the HFD system reverts control to the driver. If the enableHFDS() function is true then the HFD system is started. The HFD system issues commands to the actuators to control the vehicle. If there is a fault with the HFD system HFDS reverts control to the driver. Also shown is when there is a driver override on one of the actuators. This results in HFDS being disabled and control being reverted back to the driver.

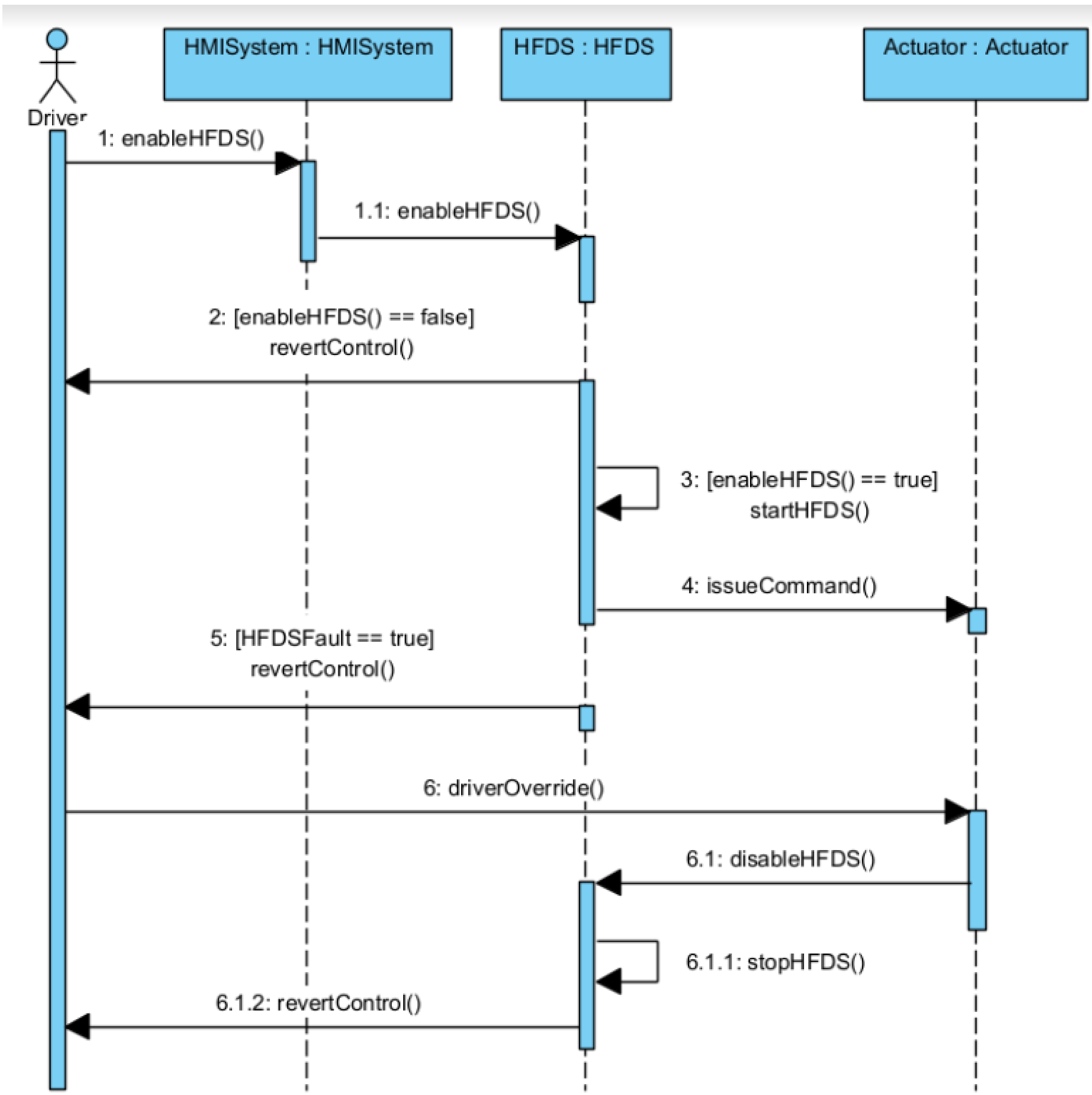

Figure 7. Sequence Diagram for Continuous Control Change

4.3.5 System Security

The following sequence diagram displays what measures can be taken to ensure only authorized commands are sent to the VehicleControlSystem. First, under normal conditions a command is sent from the DriverAssistSystem to the VehicleControlSystem. The VehicleControlSystem then gets the blue path and checks to see if there is a conflict between the path and the command. If this conflict is true, the VehicleControlSystem receives a command from the DriverAssistSystem. The VehicleControlSystem then checks the commands against each other. If there is no conflict, and the commands were checked the control system can send actions to the actuator. If there is a conflict or the commands were not checked, the actuator is sent a halt command.

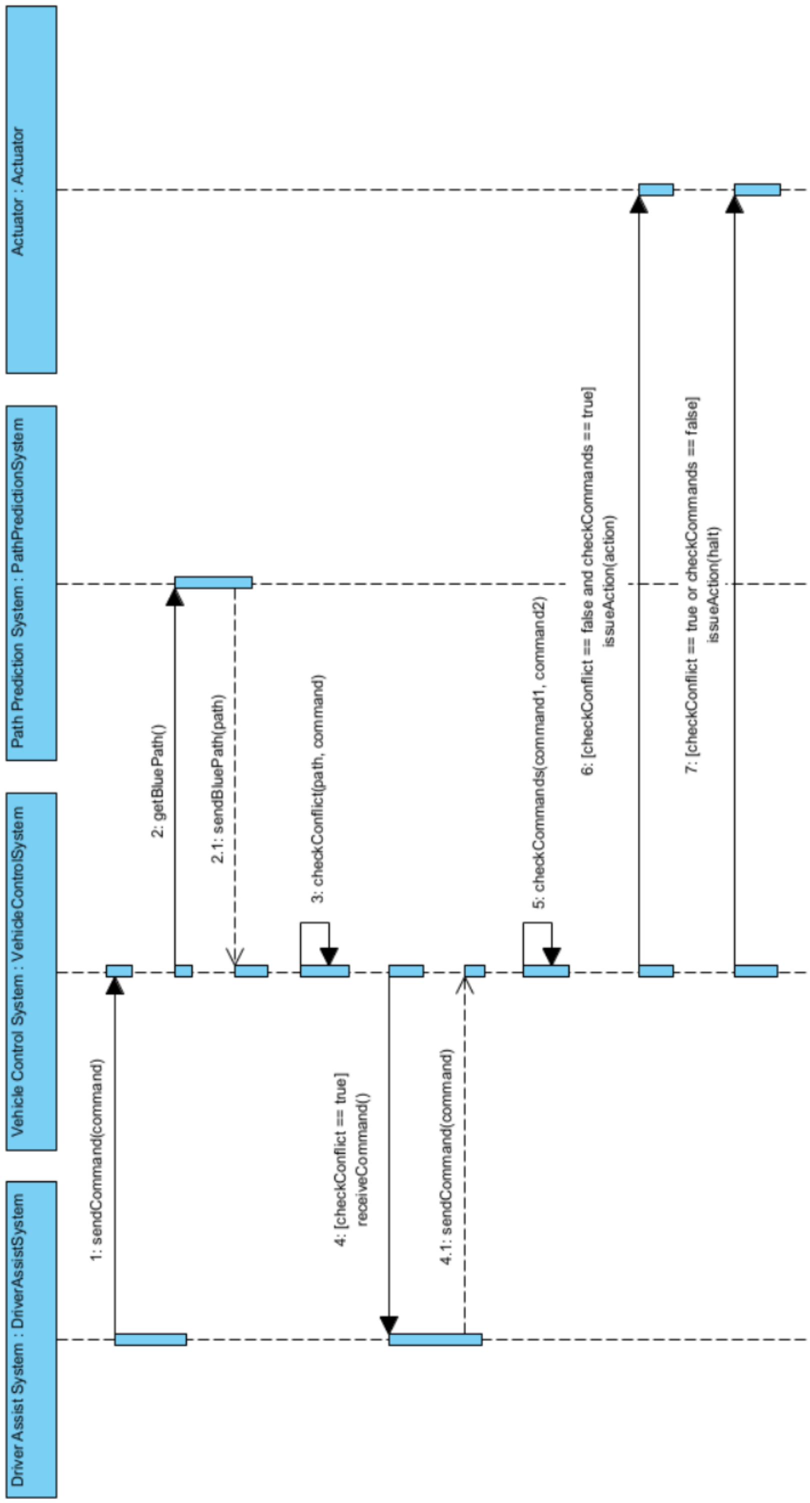

Figure 8. Sequence Diagram for Security Measures

4.4 State Diagram

This section contains the state diagram that shows how the system changes to different states when enabling HFDS. States are represented by the blue rectangles and transitions between them are represented by the named arrows. The name on the arrow represents what event took place to cause the state to change.

In the following state diagram the process by which the HFDS turns on is shown. The initial state is HFDS being disabled. If the user intiaizalies the HFD system the first procedure which is followed is the surroundings being checked. If the surroundings are deemed not safe, HFDS is not allowed to engage. If the surroundings are deemed safe, the system then moves into the next procedure of checking driver attention. If the driver is not attentive HFDS is not allowed to engage. If the driver is deemed attentive then the system moves into its next procedure of checking if the cruise control is enabled. If the cruise control is not enabled HFDS is not allowed to engage. If cruise control is enabled then the system begins vehicle position validation. If the position is invalid, HFDS is not allowed to engage. If this state is valid, HFDS is finally allowed to engage.

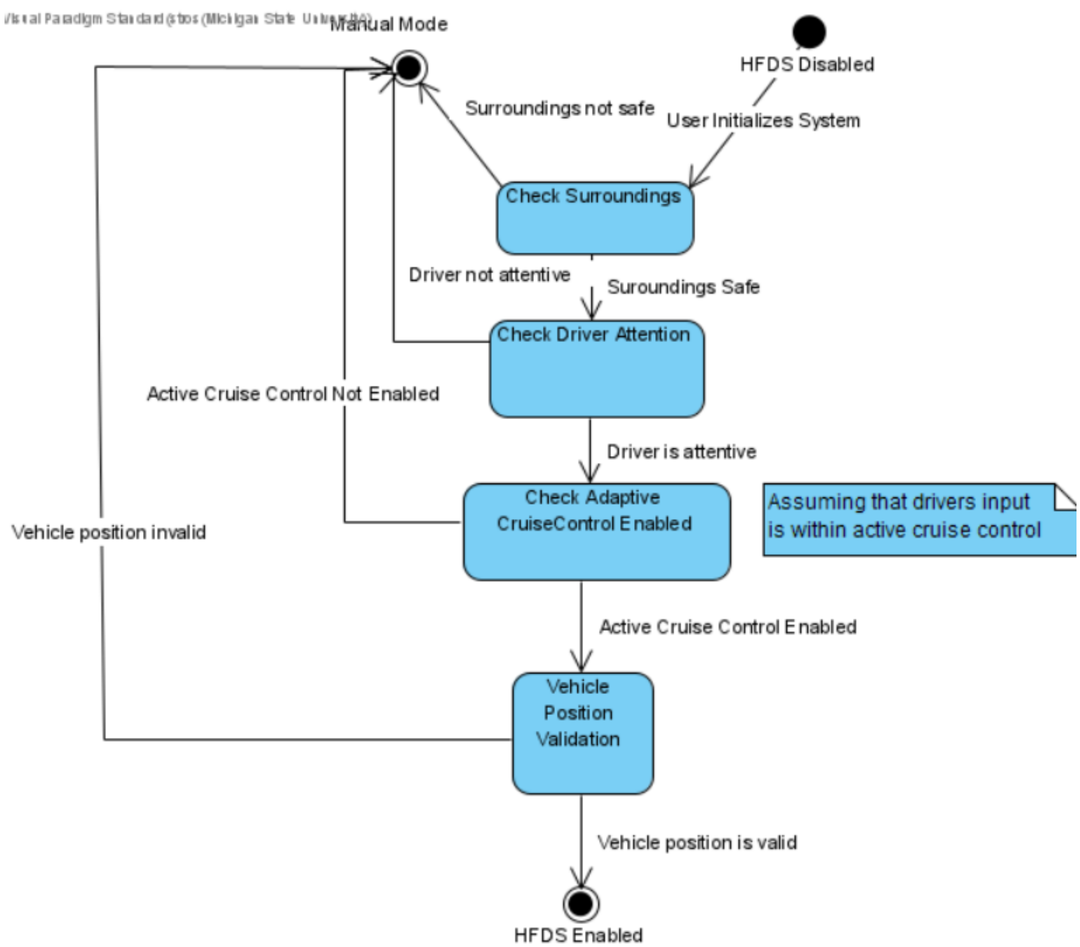

Figure 9. State Diagram for HFDS

5 Prototype V1

To visualize the HFDS that will be developed, a web demo was made that can account for elements outlined in the design documents. The first prototype demonstrates the basic activation conditions for the HFDS.

Figure 10. Web demo of Prototype

5.1 How to Run Prototype

The prototype is accessible through this link: https://cse.msu.edu/~zhouti17/HFDS4/prototypes/prototype1.html.

The Phaser.js library was used to create the prototype, it has the function to switch between WebGL rendering and canvas rendering depending on the browser used to view the webpage. As long as the browser used supports those types of rendering, the demo will run. Performance will depend on the viewer's hardware. As the controls are performed using a keyboard, one will be required to fully test the demo.

5.1.1 Controls

The prototype v1 starts with the car in a stopped state. It can be controlled using 5 keyboard keys:

- D - Drive. Currently this function instantly accelerates the car to the speed required to activate the HFDS. Gradual acceleration and speed differentiation will be added in v2.
- B - Brake. As with Drive, this function currently instantly stops the car and disables the HFDS.
- H - Enable HFDS. This function activates the HFDS only when certain conditions are met.
- F - Fault. This simulates a generic fault occurring in the HFDS. For v2, specific faults will be selectable.
- R - Restart. This does reset the prototype for now but isn't too useful. The function is more for when collision detection and steering is implemented and crashes can happen.

5.2 Sample Scenarios

The prototype currently has 4 states:

1. Car stopped
2. Car driving with HFDS off
3. Car driving with HFDS on
4. HFDS faulted

This creates the following scenarios, represented by a text output on the central console:

1. The car is stopped and the user attempts to activate HFDS, in which case the screen will prompt the user to speed up

Figure 11. Console display prompting the user to speed up

2. The car is moving and the user attempts to activate the HFDS, so HFDS turns on

Figure 12. Console display showing that HFDS has been activated

3. The car is moving and HFDS is turned on, the user activates brakes which stops the car and turns off HFDS

Figure 13. Console display showing that HFDS has been deactivated

4. The car is moving and HFDS is turned on, HFDS faults, the car continues moving while checking for driver input. Upon receiving no driver input, the car halts.

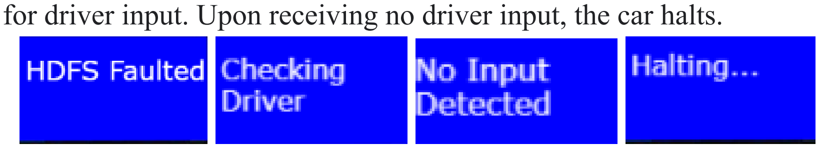

Figure 14. Sequence of console displays when the HFDS faults

6 References

[1] I. D. Brown, “Driving fatigue,” *Endeavour*, vol. 6, no. 2, pp. 83–90, 1982.

[2] P. Thiffault and J. Bergeron, “Monotony of road environment and driver fatigue: A simulator study,” *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 381–391, 2003.

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

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.