Requirements Analysis Document:
Vehicle Subsystem

JAMES Project
15-413 Software Engineering
Fall 1997
Carnegie Mellon University
Pittsburgh, PA 15213

Revision History:
Version 1.3 - Updated 11/12/97

Preface:
This document addresses the requirements of the JAMES system. The intended audience for this document are the designers and the clients of the project.

Target Audience:
Client, Developers

JAMES Members:
Gordon Cheng, Li-Lun Cheng, Christopher Chiappa, Arjun Cholkar, Uhyon Chung, Aveek Datta, John Doe, Phillip Ezolt, Eric Farng, William Ferry, Sang Won Ham, Kevin Hamlen, Pradip Hari,
1.0 General Goals
In order for the four subsystems to be able to test and evaluate their systems a vehicle simulator has to be set up as close to the real vehicle as possible. The goal of the Vehicle Subsystem is to bridge two vehicle simulators, that being the AIM/CANalyzer and the SA/RT, and provide the appropriate interface to them so the subsystem groups may effectively run and test their systems.

2.0 Current System
The current simulators from Daimler-Benz consist of a Windows SA/RT simulator written in C, that simulates only the vehicle seat. There is also the AIM/CANalyzer simulator software written in a Canalyzer language similar to C, from Daimler-Benz that can simulate multiple vehicle functions running on a laptop computer.

3.0 Proposed System

3.1 Overview
The vehicle subsystem will receive data from the other subsystems from a smart card or their on board computer. The vehicle subsystem will provide the interface for the subsystems to communicate with the selected vehicle simulator and then send back to the subsystems a response either simulating the behavior of a vehicle component or providing it with the requested information.

3.2 Functional Requirements
The vehicle subsystem’s foremost requirement is to provide access and functionality to Daimler-Benz Car simulator provided by the Daimler Benz... In addition to this functionality, the vehicle subsystem will provide platform-independent Interface and web-based access to the simulator. The integration of a seat positioning application will be demonstrated to show that a JAMES application can be integrated with the existing electronic systems in the car.

3.3 Non-functional Requirements

3.3.1 User Interface and Human Factors
No end users will use the vehicle subsystem, only other subsystems. The other subsystems do not directly access the simulator but are provided an Interface through which they can access one of two simulators through a bridge pattern that is transparent to the other subsystems and end user.

3.3.2 Documentation
Documentation on the SA/RT, AIM, Canalyzer, and the corresponding interfaces are detailed through modeling diagrams and text descriptions. The audience being addressed will be the subsystem groups and the client.

3.3.3 Hardware Consideration
A high performance processor is needed capable of running java byte code natively or interpreted. The fact that java is a cross platform language allows a good amount of flexibility here. Although there are some restraints because of the limited memory on the smart card, the processor itself does need to be fast because the subsystems are running java code.

3.3.4 Performance Characteristics
The system need not be fast, but it must be responsive. For example when the user requests their seat to be moved the seat should start moving as soon as possible. Components such as the speedometer should always be updated by the vehicle subsystem. For the end user response times should be as fast as possible. For the mechanic or dealer the times can be a bit slower but should still be reasonably fast and there should be a way for them to see what is going on and for general troubleshooting. However, on the smart card we are quite constrained by capacity, but there will be data storage available on the vehicle's onboard computer where memory space should not be as much of an issue.

3.3.5 Error Handling and Extreme Conditions
The system would prioritize the errors based on whether it has a critical effect on the other subsystems. In addition
the connection modes (coupled, loose coupled, sneaker net, and decoupled), would be taken into consideration. Only a limited number of repeated requests to the Electronic Control Units (ECU), would be executed before a simple error message is returned. This procedure is used so that the bus will not be overloaded with requests which are no longer valid. The system will also notify the maintenance subsystem of errors. Extreme conditions would just result in the return of an error message if possible.

3.3.6 System Interfacing
The vehicle subsystem will provide all interface between the other subsystems and the simulator. This will be accomplished using Java Remote Method Invocation (RMI) to create the proper bridges, adapters, and facades. Data and information will be communicated between the vehicle subsystem and other subsystems via notification which allows both pushing and pulling through different event channels. The simulator may also receive input and output through an independent web-based interface to the vehicle simulator. The hardware interface consists of a serial RS232 cable to connect the Smart Card Reader to the smart cards a GPS (global positioning system) that the vehicle will be receiving location and heading information from.

3.3.7 Quality Issues
Error checking and recoverability should be done in such a way that the system should trap all faults and try to correct problems on the fly. Only as a last resort should the user be informed of the problem. The JAMES system should remain operational throughout all driving periods.

3.3.8 System Modifications
Initially the only component that will be simulated will be the seat. It will be simulated first through a Java stub and then through the AIM/Canalyzer or SA/rt. If time permits this would be expanded to other vehicle components. The implementation side of the vehicle bridge is likely to get changed as new seat, mirror, entertainment, etc. units are introduced. This may involve simply new CAN bus ID's, or entirely new transactions on the bus. New units may also require modifications to the server side, for example if there are new parameters that must be set or read from the unit. These are not as likely as modifications to the implementation side.

3.3.9 Physical Environment
The target equipment will be operating in all Daimler Benz vehicles equipped with JAMES. The target equipment in the car will be mobile whereas the Daimler-Benz central database where the JAMES applications may store information, will be outside the vehicle. Client terminals will be distributed to all Daimler-Benz dealers. The GPS system on the vehicle is dependent on the vehicle being within range of the global positioning satellite. The mobile telecommunications of the vehicle will vary depending on the standard used (i.e. GSM in Europe and PCS in US).

3.3.10 Security Issues
The vehicle provides a concept of user access levels which will be enforced by the vehicle subsystem. These levels may specify that a user does not have access to certain data or even certain parts of the system. Likewise, access to parts of the system will be restricted based on the current operation mode of the vehicle, (off, start, idle, run, accessory). At this time we do not foresee any data that cannot be shown to all users, but depending on the user access level and vehicle operation mode certain data may be protected from being changed. Some of this data will likely be stored on the Smart Card. In this case the security methods on the card should prevent the user from tampering with the data on the card. However, smart card security is not an issue for the vehicle subsystem. The vehicle subsystem cannot protect the vehicle and the data included within from getting stolen. It is the responsibility of the owner to protect their smart card.

3.3.11 Resource Issues
There are three kinds of data that should be installed in a vehicle. The first data is engineering features of the vehicle. This has to be installed by the manufacturer and should only be maintained by licensed dealers. For example, if some parts are replaced and the main control system should be aware of it, the data should be updated accordingly. The second data is the condition of the vehicle regarding maintenance. This should be maintained by the vehicle itself as checking its condition and finding any problem. This data should be downloadable to a card so that it's given to the maintenance department. The third data is user preference data. This data should be backed up by the vehicle for default user preference setting, and a driver should also be able to backup this data in either the card or vehicle as he/she wishes.

3.4 Constraints
The vehicle subsystem is constrained by the limitations of using a Linux and NT environment. In addition, the legacy systems can not be changed and therefore restricts the use of the existing code. New source code has to
comply with the limits of the smart card and the vehicle subsystem must rely on the client for simulator documentation.

3.5 System Model

3.5.0.1 Forecast
The vehicle subsystem provides other subsystems with the interface services to the vehicle. In terms of the services it provides to other subsystems, the vehicle subsystem is divided into four areas. They are services to Logbook for getting odometer, clock, date, location readings, Maintenance for diagnostic testing, Travel for getting GPS information and VIP subsystem for controlling and getting information about the components of the vehicle.

3.5.1 Scenarios

Simulation-Logbook Scenarios
1) Scenario: Starting the vehicle
   It is early Monday morning and Bob intends to drive to work early to avoid the rush hour traffic. Bob fishes out his car keys in the garage. He unlocks the car door and inserts his key into the ignition and starts the vehicle. The processor for the vehicle starts and receives the StartVehicle signal. The vehicle gets the odometer, clock, date, and location readings and pushes the initial readings to the meters' event channels. The Logbook cardlet receives the readings from the event channels. Bob is not aware of all this but is happy when he hears the engine start running.

2) Scenario: Powering down the vehicle
   Bob guides his M-class effortlessly into the garage. He the turns his key to power down the vehicle. This causes the processor to start its PowerDown sequence. The Vehicle receives the signal and gets the meter readings. The vehicle pushes the final readings to the meters' event channels. The Logbook cardlet receives the readings from the event channels. The vehicle then shuts down all the engine of the vehicle. Bob removes the key from the ignition and steps out of the car, closes the door, locks the vehicle and walks into the house.

Simulation-Maintenance Scenarios
1) Scenario: Performing diagnostic test
   On the way home last night, Bob feels that the temperature of his car was uncomfortably higher than normal. After making a phone call to his friendly neighborhood Daimler-Benz dealer this morning, Bob drove his vehicle in for an inspection. Bob explains to Moe, the mechanic at the dealership about the problem. Moe starts his inspection by performing a diagnostic test of the vehicle. The test results indicate that a leakage is detected in the air conditioner system. Moe informs Bob of the result and quotes the repair fees. Bob scratches his head and wonders if he can do without air conditioning until he gets his new pay check at the end of the month.

Simulation-Travel Scenarios
1) Scenario: Viewing the trip
   After he fixed the air conditioner and obtained a well earned one week vacation, Bob decides to take a leisurely drive to the Grand Canyon. On the way there, Bob passes by a row of casinos. He decides to check with the traveling assistant on his M-Class vehicle if he is anywhere near Las Vegas. Bob selects an option on the traveling assistant to view his current location. The travel subsystem (cardlet) requests current position and map information from GPS. A few milliseconds later, the orbiting satellites send the exact coordinates back to the vehicle. The travel subsystem takes the coordinates and determines that Bob is indeed in Las Vegas and informs Bob of the abounding opportunities to recoup his air conditioner repair costs. Viva Las Vegas!

2) Scenario: Notification of the road/traffic/weather conditions that may affect trip
   Bob happily waits for the valet to fetch M-Class after getting lucky at the Craps table in the MGM Grand. When the vehicle arrives, Bob tosses a $50 chip to the valet as tip. The valet happily accepts the tip and informs Bob that a snow storm is expected later on in the evening. Bob wonders if he has to postpone his trip to the Grand Canyon in view of this. Once he is in his vehicle, Bob selects the option on the Travel assistant to see if the snow storm would affect the
rest of his planned trip. The travel assistant then checks for weather conditions on the routes that Bob planned. The checks revealed that a snow storm would be coming in from the north and would cross the scenic route planned by Bob. The travel assistant informs Bob of this and suggests alternative routes to reach Grand Canyon that will avoid the snow storm.

Simulation-VIP Scenarios
1) Scenario: Adjusting fuel economy
   Bob is driving down an unfamiliar stretch of highway. He is low on fuel and is not sure where the next gas station is. Bob uses the control panel to switch from normal drive to fuel economy drive so as to increase his chances of reaching the next gas station. VIP gets the change signal from the user informs the vehicle of the change in fuel economy level. Vehicle changes the mode accordingly and displays it on the dashboard. Bob looks at the new mode and feels a little more secure.

2) Scenario: Displaying cockpit controls
   Bob is driving down the same unfamiliar stretch of highway. There have been no speed signs for a while. Bob decides to find out how far he has traveled and at what speed. He requests the information from VIP. VIP informs the vehicle of the requests, which then fetches the information from the speedometer and odometer and displays it to the user via the dashboard. Bob looks at the information and feels a little more secure again.

3) Scenario: Adjusting cockpit controls
   Bob is still driving down the same unfamiliar stretch of highway. He wants to make sure that a state trooper does not catch him for speeding. He informs VIP that he wants to set his speed limit at 50 mph. Bob also wants to see how far it will be from his starting point to the next gas station. He informs VIP which then relays the message to the Vehicle to do the calculation. Vehicle will then wait till Bob reaches the gas station and display the relevant data to him.

4) Scenario: Adjusting steering wheel
   Bob feels a strain in both his hands and would like to adjust the steering wheel to a more comfortable position. Bob enters the desired x and y coordinates to VIP. VIP conveys the message to Vehicle, which then adjusts the steering wheel completely. Bob feels comfortable now.

5) Scenario: Initial Query
   Bob wants to find out what special components he has in his new M-Class. He sends the request to VIP, which then gets the relevant available components and displays it to Bob.

6) Scenario: Adjusting the seat
   Bob feels uncomfortable and wants to adjust his seat "Duke of Hazard" style. He adjusts his seat via the VIP, which then sends the desired parameters of the new seat position to the Vehicle, which then adjust the seat mechanically. Bob feels confident and comfortable after the seat adjustment.

7) Scenario: Adjusting the mirror
   Bob accidentally knocks the rearview mirror off the original position. He informs the VIP to reset his mirror back to the original position. VIP gets the original parameters and sends them to Vehicle. Vehicle uses the information to mechanically adjust the mirror back to its original position. Bob looks at the mirror again and sees a trooper behind his car!

8) Scenario: Adjusting the radio
   Bob is feeling tired driving. He wants to change tunes from country to heavy metal. He knows which local station plays the best heavy industrial metal. Bob says "WRAD" to VIP. VIP picks up this request and changes the radio channel to WRAD. Bob feels at ease when he hears Nine-Inch Nail playing over the airwaves.

9) Scenario: Adjusting the air conditioner
   Bob feels hot while listening to heavy metal. He decides that he should cool down by reducing the temperature in his vehicle. He says "colder" to the VIP. VIP picks up the signal and adjusts the thermostat of the air conditioner to lower the temperature in the vehicle.
3.5.2 Use Case Models

3.5.2 Actors

User - Driver or administrator of the car.
Vehicle Subsystem - System that provides other subsystems with interfaces to, and simulation of, the components of the vehicle.
Logbook Subsystem - System that provides functionalities for logbook keeping of the vehicle including trip/leg information and user access rights.
Travel subsystem - System that provides functionalities to aid the driver with his/her traveling.
Maintenance subsystem - System that provides information and functionalities regarding maintenance and diagnostics of the vehicle.
VIP subsystem - System which controls the settings of the vehicle's functional components.

3.5.2.2 Use Cases

The top level use case illustrates the relationship between the various subsystems and the vehicle. The vehicle subsystem provides the other subsystems with interface services to the vehicle. In terms of the services it provides to other subsystems, the vehicle subsystem is divided into four areas. They are services to Logbook for getting odometer, clock, date, and location readings. Maintenance for diagnostic testing. Travel for getting GPS information, and VIP subsystems for controlling and getting information about components of the vehicle.

FIGURE 3.5.2.1. Top level Vehicle use cases.
FIGURE 3.5.2.2. Logbook-Simulation Use Cases

Use Case: Get Odometer
Actors: Vehicle, Logbook, User
Entry Condition: User gives a StartVehicle or PowerDown event.
Flow of Events:
1. Vehicle receives a StartVehicle or PowerDown event from User.
2. Vehicle reads the odometer.
3. Vehicle pushes the initial or final odometer reading to the meter's event channel.
4. Logbook gets the reading from the channel.
Exit Condition: Logbook receives the signal.
Special Requirements:
1. Vehicle must be able to delay power down after the user requests it for long enough to execute all power down
   sequences, such as passing the odometer reading to the all interested listeners.
2. Logbook must be subscribed to the meter's event channel.
Exceptions:
1. The vehicle cannot read the odometer: Notify maintenance of problem, and pass error signal to meter's event
   channel.
2. Logbook does not receive signal: Notify maintenance of problem.

Use case Name: Get Time
Actors: Vehicle, Logbook, User
Entry Condition: User gives a StartVehicle or PowerDown event.
Flow of Events:
1. Vehicle receives a StartVehicle or PowerDown event from the User.
2. Vehicle reads the clock.
3. Vehicle pushes the initial or final clock reading to the meter's event channel.
4. Logbook gets the reading from the channel.
Exit Condition: Logbook receives the signal.
Special Requirements:
1. Vehicle must be able to delay power down after the user requests it for long enough to execute all power down sequences, such as passing the clock reading to the all interested listeners.
2. Logbook must be subscribed to the meter's event channel.
Exceptions:
1. The vehicle cannot read the clock: Notify maintenance of problem, and pass error signal to meter's event channel.
2. Logbook does not receive signal: Notify maintenance of problem.

Use case Name: Get Location
Actors: Vehicle, Logbook, User
Entry Condition: User gives a StartVehicle or PowerDown.
Flow of Events:
1. Vehicle receives a StartVehicle or PowerDown event from the User.
2. Vehicle reads the location.
3. Vehicle pushes the initial or final location reading to the meter's event channel.
4. Logbook gets the reading from the channel.
Exit Condition: Logbook receives the signal.
Special Requirements:
1. Vehicle must be able to delay power down after the user requests it for long enough to execute all power down sequences, such as passing the location reading to the all interested listeners.
2. Logbook must be subscribed to the meter's event channel.
Exceptions:
1. The vehicle cannot read the location: Notify maintenance of problem, and pass error signal to meter's event channel.
2. Logbook does not receive signal: Notify maintenance of problem.

FIGURE 3.5.2.3. Maintenance-Simulation Use Cases
Use Case Name: GetDiagnostics
Actors: Maintenance, Vehicle
Entry Condition: Maintenance sends a GetDiagnostics request to Vehicle.
Flow of Events:
1. Vehicle receives a GetDiagnostics request from Maintenance.
2. Vehicle gets the odometer (mileage/km) reading, the engine reading, the error log, current date, last service date, and other diagnostic information.
3. Vehicle returns the information to Maintenance.
Exit Condition:
Maintenance acknowledges the diagnostic.
Special Requirements:
The vehicle must have ECUs able to give status info on the parts.
Exceptions:
The vehicle will return a composite error message after it has checked for all the different readings if any components are missing.

FIGURE 3.5.2.4. Travel-Simulation Use Cases

Use case Name: Get Position.
Actors: Vehicle, Travel, Driver
Entry Condition: Driver selects a function from travel application, that needs current position information.
Flow of Events:
1. Travel sends a message to vehicle to get position information.
2. Vehicle receives the message and sends the request to GPS.
3. GPS sends the information to the vehicle.
4. Vehicle sends the information to the travel.
5. Travel receives the information.
Exit Condition: Travel receives the information.
Special Requirements:
1. GPS response time should be short enough to give the driver accurate information when needed.
Exceptions:
1. The vehicle cannot communicate with GPS server.
2. GPS server is not available.
Use case Name: Get routes.
Actors: Vehicle, Travel.
Entry Condition: A driver selects a function from travel application to see alternative routes.
Flow of Events:
1. Travel sends the request to vehicle to get alternative routes information.
2. Vehicle sends the request to GSM.
3. GSM sends the information to vehicle.
4. Vehicle receives the information and sends it to travel.
5. Travel receives the information.
Exit Condition: Travel receives the alternative routes information.

Special Requirements:
1. GSM sends the condition information quickly enough so that the driver has enough time to respond to the situation.

Exceptions:
1. The vehicle cannot communicate with GSM server.
2. GSM server is not available.

Use case Name: subscribe_to_GSM
Actors: Vehicle, Travel, Driver.
Entry Condition: A driver inserts a card into card reader.
Flow of Events:
1. Travel reads trip data from the card.
2. Travel sends subscription request to vehicle.
3. Vehicle receives the request and send it to GSM.
4. GSM receives the request and acknowledges the vehicle.
Exit Condition: Vehicle receives the acknowledge from GSM.

Special Requirements:
1. GSM sends the condition information quickly enough so that the driver has enough time to respond to the situation.

Exceptions:
1. The vehicle cannot communicate with GSM server.
2. GSM server is not available.

Use case Name: Notification of the conditions that may affect the trip.
Actors: Vehicle, Travel, Driver, GSM.
Entry Condition: GSM notifies the vehicle of any conditions that may affect the route planned.
Flow of Events:
1. Vehicle receives the notification from the GSM.
2. Vehicle sends the information to the travel.
3. Travel receives the information.
Exit Condition: Travel receives the condition information.

Special Requirements:
1. GSM sends the condition information quickly enough so that the driver has enough time to respond to the situation.

Exceptions:
1. The vehicle cannot communicate with GSM server.
2. GSM server is not available.
FIGURE 3.5.2.5. VIP-Simulation Use Cases: Drive Controls

Use-case name: ADJUST FUEL ECONOMY
Actors: VIP, Vehicle and Driver
Entry Condition: Driver sends new mode of fuel economy to VIP
Flow of events: VIP sends message to Vehicle to change current fuel economy mode
Vehicle adjust the mode according to the instructions from VIP
Vehicle informs VIP of the current odometer and fuel supply.
Exit Condition: VIP acknowledges the change
Exceptions: The fuel gauge breaks or the display on the dashboard breaks down, VIP and Maintenance is informed of the defect.

Use-case name: ADJUST COCKPIT CONTROLS
Actors: VIP, Vehicle and Driver
Entry Condition: Driver sends desired speed/mileage level to VIP
Flow of events: VIP sends message to Vehicle to set the speed/mileage level
Vehicle adjust to the new speed/mileage
Vehicle informs VIP of the changes and displays the new levels of speed/mileage to the driver via the dashboard display
Exit Condition: Driver sees the speed/mileage level
Special Requirements: Driver cannot be allowed to exceed legal speed limits
Exceptions: Speedometer or Odometer mechanical failure, Maintenance is informed.

Use-case name: ADJUST STEERING WHEEL
Actors: VIP, Vehicle and Driver
Entry Condition: Driver sends desired horizontal and vertical position of the steering wheel to VIP
Flow of events: VIP sends the horizontal and vertical coordinates to Vehicle
Vehicle mechanically adjusts the steering wheel
VIP is informed of new levels
Exit Condition: Driver "feels" the new position of the steering wheel
Special requirements: the vehicle should not be moving, i.e. its mode of operations should be halt.
Exceptions: Steering wheel suffers from mechanical failure, maintenance is informed.
The vehicle is moving and returns a message to VIP to refuse the adjustment.

ADJUST CONVENIENCE CONTROLS

FIGURE 3.5.2.6. VIP-Simulation Use Cases: Convenience Controls

Use-case Name: GET_COMPONENT_LIST
Actors: The VIP subsystem and the Vehicle subsystem.
Enter condition: The VIP subsystem sends a message to the vehicle subsystem to ask for the available components.
Exit condition: The VIP subsystem sends a message to the vehicle subsystem to adjust the seat, as well as the required parameters.

NOTE: The following use cases represent instances of the GET_component_Value and Adjust_component.

Use-case Name: ADJUST SEAT
Actors: VIP and Vehicle.
Enter condition: The VIP subsystem sends a message to get the position of the seat.
Flow of events: The VIP subsystem sends a message to the vehicle subsystem to adjust the seat, as well as the required parameters.
The vehicle processes the message to the seat which adjust its position according to the given parameters.
The VIP subsystem sends a message to the vehicle subsystem to adjust the seat headrest, along with the required parameters. The vehicle subsystem processes the message to the seat which adjusts the headrest accordingly. Exit condition: The vehicle sends a message to the VIP subsystem stating that the request has been executed. Special requirements: The vehicle has to have an adjustable seat. Exceptions: If the seat is broken or if the seat cannot move in the required direction, the vehicle sends a message to the VIP subsystem about the status of the seat.

Use-case Name: ADJUST MIRROR
Actors: VIP and Vehicle.
Entry condition: The VIP subsystem sends a message to get the position of the mirror.
Flow of events: The vehicle sends the position of the mirror.
The VIP subsystem sends a message to the vehicle subsystem to adjust the mirror along with the required parameters. The vehicle processes the request to the mirror, which adjusts itself according to the given parameters. Exit condition: The vehicle sends a message to the VIP subsystem stating that the request has been executed. Special requirements: The vehicle has to have an adjustable mirror. Exceptions: If the mirror is broken or if the mirror cannot move in the required direction, the vehicle sends a message to the VIP subsystem, about the status of the mirror.

Use-case Name: ADJUST RADIO
Actors: The VIP subsystem and the Vehicle subsystem.
Entry condition: The VIP subsystem sends a message to the vehicle subsystem to get the status of the car stereo.
Flow of events: The vehicle replies by sending the status of the radio.
The VIP subsystem sends a message to the vehicle subsystem to open the radio.
The vehicle processes the request to the radio to turn its power on.
The VIP subsystem sends a message to vehicle to select a radio channel.
The vehicle processes the request to the radio.
The VIP subsystem sends a message to adjust the volume (it also sends the required parameter).
The vehicle subsystem processes the request to the radio.
Exit condition: The vehicle sends a message to the VIP subsystem stating that the request has been executed. Special requirements: The vehicle has to have a radio. Exceptions: if the radio is malfunctioning for any reason the vehicle has to notify the VIP subsystem.

Use-case Name: ADJUST AIR_CONDITION
Actors: VIP and Vehicle.
Entry condition: The VIP subsystem sends a message to the vehicle subsystem to get the status of the air condition (on or off and temperature).
Flow of events: The vehicle replies by sending the status of the AC.
The VIP subsystem sends a message to the vehicle subsystem to open the air condition with the required temperature.
The vehicle processes the open and adjust requests to AIR_CONDITION.
Exit condition: The vehicle sends a message to the VIP subsystem stating that the request has been executed. Special requirements: The vehicle has to have an air condition. Exceptions: if the AC is malfunctioning for any reason the vehicle has to notify the VIP subsystem.

3.5.3 Object Models.

FIGURE 3.5.3.1 GENERAL STRUCTURE OF THE VEHICLE SUBSYSTEM
The vehicle class represents the vehicle facade by which the subsystem will communicate. It contains Components, and one OperationMode and CarDescriptor. It also contains a User class which is used for authentication purposes. The OperationMode class has dynamic behavior. It has one attribute status, the values of which can be: run/ off/ start/ idle/ accessory-battery. The CarDescriptor class contains three attribute: Configuration, (which contains a dynamic component list), Model number, and Serial number

FIGURE 3.5.3.1 - VEHICLE SUBSYSTEM CLASS DIAGRAM
All components of the vehicle are either composite components or leaf components. Examples of composite
component include the dashboard and accessories. Examples of leaf components include seat, air conditioner,
dashboard-meters, steering wheels, etc. The location and heading components which are also dashboard-meters
require an access to a GPS system which is accessed through AIM.

GENERAL OPERATIONS

The vehicle class will include the following methods:
getComponentList
gGetComponentValue
void setComponentValue
getModel
gerserial
gerUserList
addComponent
removeComponent

The component class will have methods, set and get, that are inherited by all leaf components.
The composite component will have four methods, set, get, addComponent and removeComponent. These are used
in case a component is installed or replaced at any time during the lifecycle of the vehicle. All dashboard meters will
have an additional method called display and
all leaf components will access the two different simulators by using the bridge pattern.
3.5.4 Dynamic Models

FIGURE 3.5.4.1 SHOWS TRAVEL SUBSYSTEM REQUESTING CURRENT LOCATION OF VEHICLE

NAVIGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.1:
1. The Travel subsystem requests the current position of the vehicle from the GPS.
2. The GPS checks the coordinates and returns the current position to the Travel subsystem.

FIGURE 3.5.4.2 SHOWS TRAVEL SUBSYSTEM REQUESTING ALTERNATE ROUTE
FIGURE 3.5.4.2 - TRAVEL AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR GET ROUTE

NAVIGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.2:
1. Travel subsystem sends subscription to Global Communications Systems (PCS & GSM) so as to obtain up to date route information.
2. Global Communications sends the route condition to Travel subsystem.
3. Travel subsystem determines from the route condition that an alternative route is needed and requests further information from Global Communications Systems.
4. Global Communications System sends a new route information to Travel subsystem.

FIGURE 3.5.4.3 SHOWS VIP SUBSYSTEM ADJUSTING TO FUEL ECONOMY MODE.
FIGURE 3.5.4.3 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR ADJUST FUEL ECONOMY

NAUTIGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.3:
1. Driver sends a new mode of fuel economy via VIP.
2. VIP notifies the vehicle of the new fuel economy mode.
3. Vehicle updates its fuel economy mode.
5. VIP displays the new fuel economy mode via the dashboard display.

FIGURE 3.5.4.4 SHOWS HOW VIP SUBSYSTEM DISPLAYS THE COCKPIT CONTROLS.
FIGURE 3.5.4.4 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR DISPLAY COCKPIT CONTROLS

NAVI GATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.4:
1. Driver requests speed/mileage information from VIP.
2. VIP forwards the requests to vehicle.
3. Vehicle checks for the current speed/mileage.
4. Vehicle sends the information to VIP which displays the information.

FIGURE 3.5.4.5 SHOWS VIP SUBSYSTEM ADJUSTING COCKPIT CONTROLS
FIGURE 3.5.4.5 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR ADJUST COCKPIT CONTROL

NAVICATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.5:
1. Driver sets the new speed/mileage through VIP.
2. VIP forwards the new speed/mileage to vehicle.
3. Vehicle adjusts to the new speed/mileage accordingly.
4. Vehicle informs VIP of the updates.
5. VIP displays the new level of the speed/mileage via the dashboard.

FIGURE 3.5.4.6 SHOWS HOW VIP SUBSYSTEM GETS THE COMPONENT LIST.
FIGURE 3.5.4.6 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR GET COMPONENT LIST

Navigational text for sequence diagram shown in Figure 3.5.4.6:

1. VIP requests the available accessories.
2. Vehicle returns the list of identifiers for the available accessories.

FIGURE 3.5.4.7 SHOWS VIP SUBSYSTEM ADJUSTING THE STEERING WHEEL.
NAVIGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.7:
1. Driver sends the desired position of steering wheel to VIP.
2. VIP sends the horizontal and vertical coordinates to vehicle.
3. Vehicle mechanically adjusts the steering wheel.
4. Vehicle informs VIP of the new position of the steering wheel.
5. Driver "feels" the new position of the steering wheel.

FIGURE 3.5.4.8 SHOWS VIP SUBSYSTEM ADJUSTING THE SEAT.
FIGURE 3.5.4.8 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR ADJUST SEAT

NAVIGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.8:
1. VIP requests current seat position from vehicle.
2. Vehicle returns the requested information to VIP.
3. VIP requests seat adjustment from vehicle.
4. Vehicle processes the request to adjust the seat.
5. The seat adjusts its position.
6. The seat informs the vehicle of its new position.
7. Vehicle informs VIP that the request for seat position adjustment has been fulfilled.
8. VIP requests head rest adjustment from vehicle.
9. Vehicle processes the request to adjust the head rest.
10. The head rest adjusts its position.
11. The head rest informs the vehicle of its new position.
12. Vehicle informs VIP that the request for head rest position adjustment has been fulfilled.

FIGURE 3.5.4.9 SHOWS VIP SUBSYSTEM ADJUSTING THE RADIO.
FIGURE 3.5.4.9 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR ADJUST RADIO

NAVGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.9:
1. VIP requests radio status from vehicle.
2. Vehicle returns radio status to VIP.
3. VIP requests turning the power of the radio on.
4. Vehicle processes the request.
5. Radio turns on.
6. Vehicle informs VIP that the radio has been turned on.
7. VIP requests for volume adjustment.
8. Vehicle processes the request.
10. Vehicle informs VIP that the radio volume has been adjusted.
11. VIP requests for channel change.
12. Vehicle processes the request.
13. Vehicle informs VIP that the radio channel has been changed.

FIGURE 3.5.4.10 SHOWS VIP SUBSYSTEM ADJUSTING THE AIR CONDITIONER.

ADJUST AIR CONDITION

VIP

Vehicle

Air condition

1: request AC status

2: return AC status

3: requests AC adjustments

4: process request

5: adjust temperature

6: informs VIP that request is done

FIGURE 3.5.4.10 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR ADJUST AIR CONDITION

NAVIGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.10:
1. VIP requests air conditioner status from vehicle.
2. Vehicle returns air conditioner status.
3. VIP requests air conditioner adjustments.
4. Vehicle processes the request.
5. Air conditioner adjusts the temperature accordingly.
6. Vehicle informs VIP that the request has been carried out.

FIGURE 3.5.4.11 SHOWS VIP SUBSYSTEM ADJUSTING THE MIRROR.
FIGURE 3.5.4.11 - VIP AND VEHICLE SUBSYSTEM SEQUENCE DIAGRAM FOR ADJUST MIRROR

NAVIGATIONAL TEXT FOR SEQUENCE DIAGRAM SHOWN IN FIGURE 3.5.4.11:
1. VIP requests mirror position from vehicle.
2. Vehicle returns the mirror position to VIP.
3. VIP requests a mirror adjustment.
4. Vehicle processes the request.
5. Mirror adjusts its position.
6. Vehicle informs VIP that the request has been carried out.

3.5.4.1 State Charts

FIGURE 3.5.4.1.1 - STATE CHART FOR OBJECT OPERATIONMODE
The start state is when the engine is running and in parking gear.

The idle state is when the vehicle is in drive gear but stopped.

The run state is when the vehicle is in drive gear but moving.

The accessory-battery state is when the engine is running just to operate accessories.

The off state is when the vehicle has power down. The off state is both the initial state and end state of the vehicle.