Coursework – Driver' Assistance

Davide Pollicino -40401270

SET10112
[1] Edinburgh Napier University, 10 Colinton Road, Edinburgh, EH10 5DT

Abstract. This paper introduces and describes the design, infrastructure and implementation of a car 'assistant realized by using ADA Spark. The introduction section provides and description about the minimum project requirements and list of records / attributes used to implements them. The proof section is dedicated to the to presentation of a full of concept of one of the implement procedures, which details will be given in the following sections. Finally, it will be presented and described the 'extension' section, which contains a set of extra requirements implemented to enrich further the car' assistant and offer more functionality, compared to the first basic question requested.

Keywords: Ada, Spark, Driver' Assistance, Formal Approaches to software Engineer

1 Introduction

In the next paragraph it will be described the design and development of a prototype version of a car driver' assistance using ADA Spark. The system was developed using the following rules that were provided by the project specification:

1. The car cannot be turned on unless it is in Parked.
2. The car cannot be driven unless there is a minimum charge in the battery.
3. Once in motion, the system will warn of low charge.
4. The speed limit can never be exceeded.
5. The speed of the car must be zero in order to change gear.
6. If the car's sensor detect an object, then the car cannot move towards that object.
7. The car must have a diagnostic mode with renders it incapable of any other operator.

A part from the set of requirements indicated above, the implemented prototype offers an extension of requirements which will be presented in the next paragraph of this document.

2 Controller structure

The car' assistance prototype implemented has been made thinking about a tesla Car, a car powered by electric powered with higher performances and integrated driver' assistance.

The following variables has been used to implement the given prototype: (here add a table of the variables)

Variable Name	Value Domain	Description
PowerLevel	(On, off)	Describe car' system activity status (On or off).
BatteryLevel	range 0..100	Contains the battery charge status of the car
BatteryDegradationlevel	range 0..100	Value that indicated the degradation level of the battery charge, which may change in according to speed and number of passengers
CarSpeed	range 0..100	Current car' speed
GearInserted	range -1..5	Current car' gear inserted. 0 Is neutral, -1 is reverse.
MaintenaceMode	(On, off)	Maintenance mode of the car. If enabled, the car cannot move.
NumberOfPassengers	range 0..5	Number of passengers inside the car
ObjectDetected	(on, Off)	Used to
MinimumBattery	range(0..100)	Minimum battery charge needed to keep the system running
Parking	(On, Off)	Allows to active or deactivate parking mode (which switch the classic hand brake)
SpeedLimit range	(0..100)	Current road speed limit

The car'assistance controller is developed into three core files:

- levels.ads
- levels.adb
- main.adb

The main.adb file contains the contact point between the business logic of the prototype and the final user. The main file is the core file that gets executed when the ada spark project is build and executed. Within this file it has been implemented a body task that gets executed when the project is build. Within the body task it has been implement a loop used for the management and testing of the car driving assistance.

At each iteration, the program checks if the engine is on and current number of passengers. In according to the graduation level, if the engine is running, the battery charge decrease over the time. Also at each iteration, until the user does not click type any non-supported key to stop the application, the system will be in hold, available for for user instructions.

A part from a set of minimum requirements indicated in the previous paragraph, a set of extra features have been developed and will be discussed in the 'extensions' section.

3 Descriptions of procedures and functions

This section aims to provide a more detailed look at the procedures and function implemented within the codebase.

The SPARK level achieved for each procure will be indicated in the relevant paragraph title.

3.1 TurnEngineOn Procedure (SPARK GOLD LEVEL)

This procedure allows to turn on the Engine, setting the PowerLevel to On.
The precondition is that PowerLevel is Off , current Gear is in neutral, Parking is enabled and maintenance mode is off. .
Post-condition, is that PowerLevel is on, current Gear is in neutral, Parking is enabled and maintenance mode is off.

3.2 TurnEngineOff Procedure (SPARK GOLD LEVEL)

This procedure allows to turn of the Engine, setting the PowerLevel to Off, and set the battery degradation metric to zero. The precondition is that PowerLevel is On, the post-condition, is that PowerLevel is Off and BatteryDegradationLevel is 0.

3.3 UnsetParkingMode Procedure (SPARK GOLD LEVEL)

This procedure allows to Unset the parking mode, a car configuration needed to disable the electronic handbraking, and let the driver drive the car. The precondition is that Parking is On, Car speed is zero (0), and the gear inserted is 0 (also considered as neutral). The post-condition is that Parking is Off, Car speed is zero (0), and the gear inserted is 0 (also considered as neutral).

3.4 SetParkingMode Procedure (SPARK GOLD LEVEL)

This procedure allows to set the parking mode, a car configuration needed to enable the electronic handbranking, allowing the car to be safely stationary. The precondition is that Parking is Off, Car speed is zero (0), and the gear inserted is 0 (also considered as neutral). The post-condition is that Parking is On, Car speed is zero (0), and the gear inserted is 0 (also consiered as neutral).

3.5 AddPassenger Procedure (SPARK GOLD LEVEL)

This procedure allows the car' assistance to add a passenger, being aware of the number of passengers on board, and manage the battery degradation level in according to speed and number of passengers. Our car can actually allow max 5 passengers. The precondition is that Car speed is 0 (so the car is stopped), and Number of passenger is >=0 and <=4; The post-condition is that Car speed is 0 (so the car is stopped), and Number of passenger is >=0 and <=5;

3.6 RemovePassenger Procedure (SPARK GOLD LEVEL)

This procedure allows the car' assistance to remove a passenger, being so aware of the number of passengers on board, and manage the battery degradation level in according to speed and number of passengers. Our car can actually allow max 5 passengers. The precondition is that Car speed is 0 (so the car is stopped), and Number of passenger is >=0 and <=5; The post-condition is that Car speed is 0 (so the car is stopped), and Number of passenger is >=0 and <5;

3.7 EnableDiagosticMode Procedure (SPARK GOLD LEVEL)

This procedure allows to enable diagnostic mode, a car configuration will not allow the car to do any operation, expect to print car's diagnostic information (like battery level, gear inserted, degradation status). The precondition is that DiagnosticMode is Off The post-condition is that DiagnosticMode is On.

3.8 DisableDiagosticMode Procedure (SPARK GOLD LEVEL)

This procedure allows to disable the diagnostic mode, allowing so the car'assistante to execute any implemented functionality. The precondition is that DiagnosticMode is On The post-condition is that DiagnosticMode is Off.

3.9 IncreaseSpeed Procedure (SPARK GOLD LEVEL)

This procedure allows to increase the speed by 5Km/h and increase the battery degradation level as result of an higher speed. The precondition is that Engine is On, Parking mode is Off, diagnostic Mode is Off, and Gear Inserted is at least 1 (>=1), BatteryLevel is >=5 (above the minimum level), and at least the driver is inside the car (having so N. Passenger >=1). The post-condition is that Engine is On, Parking mode is Off, diagnostic Mode is Off,and Gear

Inserted is at least 1 (>=1), BatteryLevel is >=5 (above the minimum level), and at least the driver is inside the car (having so N. Passenger >=1).

3.10 DecreaseSpeed Procedure (SPARK GOLD LEVEL)

This procedure allows to decrease the speed and decrease the battery degradation level as result of a lower speed. The speed will be decreased by 5KM/h. The precondition is that Engine is On, Parking mode is Off, diagnostic Mode is Off The post-condition is that Engine is On, Parking mode is Off, diagnostic Mode is Off.

3.11 Turn Procedure (SPARK GOLD LEVEL)

This procedure allows the car to decide if it is safe to turn or not. The car cannot turn if one of its sensor will detect an object that may interface with driver and passengers safety. If the car is relative slow (under 5), its speed will be increased in order to make a safe and relatively quickly turn.

The precondition is that the Engine is On, the Gear inserted is at least 1 (>=1), the parking Mode is Off, the DiagnosticMode is Off, and that object detection is off. The post-condition is that: Engine is On, the Gear inserted is at least 1 (>=1), the parking Mode is Off, the DiagnosticMode is Off, and that object detection is off.

4 Proof of consistency

S = Car Speed
P = Number of passengers

$$\frac{\frac{\sqcap \Rightarrow S=0 \qquad \sqcap \Rightarrow P-1>=0}{S=0,\ P>=1,\ P<=5 \Rightarrow P-1<5}}{S=0,\ P>=1,\ P<=5 > S=0 \wedge P-1=0}\ R\wedge$$

$$\frac{S=0,\ P>=1,\ P<=5 \Rightarrow S=0 \wedge P-1=0 \wedge P-1<5}{S=0 \wedge P>=1 \wedge P<-5 \Rightarrow S=0 \wedge P-1>=0 \wedge P-1<5}\ L\cap$$

$$\frac{}{(S=0 \wedge P>=1 \wedge P<=5) > (S=0 \wedge P-1>=0 \wedge P-1<5)}\ R\supset$$

5 Extensions

A part from the minimum project requirements given by the coursework specifications, a second set of requirements has been implemented. The following set of extensions has been implemented:

1. Menu management
2. Battery degradation level
3. Passengers Management
4. Car's info system
5. Charge battery mode.

5. 1 Menu Management

The access to each interaction with the car a menu has been made available and accessible to each user. The user will be able to print at any time with a menu which will allow the user to interact with the system, and obtaining car information, which details will be shown in the next paragraph.

5. 2 Battery Degradation

As we know already, the car used for the implemented prototype is an electric one.
Now, as engines powered by fuel, is necessary to have fuel which feeds the combustion and allows the engine to keep running.
In our case, is necessary to have charged batteries in order to exploit even basic cars functionalities and the car'assistant itself.

The battery degradation level is a metric of energy usage that is incremented or decremented in according to the following factors

- Engine running
- Number of passengers
- Car speed

We can so understand that as long the car speed improves, higher is the quantity of emergency necessary to maintain that speed; By adding more passengers and increase so the load (or weights) of the car, higher is effort needed to simply move the car.
At each iteration, while active, the car' assistant check if the engine is running and decrease the battery charge available in according to the battery degradation level.

Within the implemented prototype, the user will receive a message is the battery charge is low, before that the system will automatically stop, once the battery charge reaches the minimum charge level required for the system work.

5. 3 Passengers Management

Is it possible to add or remove passengers inside the car, Just when the car has speed 0. As each passenger adds weights inside the car, the battery degradation increase, decrease by consequence the battery charge lifetime.

Load passengers has the following conditions:

- Pre: Number of Passengers <=4 and NumberOfPassenger >=0 and CarSpeed =0
- Post: Number of Passengers <=4 and NumberOfPassenger >=1 and CarSpeed =0

Passengers have an impact on the battery degradation level, which is increase every time a new passenger come onboard.

The unload of a passenger has the following condition:

- Pre: Number of Passengers <=5 and NumberOfPassenger >=1 and CarSpeed =0
- Post: Number of passengers >=0 and NumberOfPassengers <=5;

As each passenger has a weigh, the battery degradation level will be decreased once a passenger leave the car.

5.4 Car' info system

Provide information related to:

1. Car speed
2. Battery level
3. Battery degradation level
4. Number of passengers
5. Engine status
6. Current Gear inserted
7. Parking mode inserted
8. Maintenance status

5.5 Battery Charge mode

As the battery charge gets exploited while driving, is necessary to recharge the Tesla car batteries. The pre-condition for enabling the charging mode is that the car speed is 0, the engine is

off, Parking mode is enabled and charging mode is off. The post condition is that Charging mode is On and car speed is zero.

6 Conclusion

The level of Spark reached at the end of the implementation and testing is: Gold.

All the minimum requirements of the project have been satisfied and extra functionalities like battery degradation management and passenger number management has been implemented.

The product itself provides also a complete UI that allows to quickly retrieve information about current car' status, having also information about Battery percentage and battery degradation level.

As part of the design, the car will not be able to overcome current road speed limit for security reasons.

The battery degradation level, as previously discussed has been implemented taking under consideration car's speed and number of passengers. Further improvements to the battery degradation of the car could be correlated to air conditioner, built-in car internet connectivity system, and asset selected (city – sport, for higher performances), respecting so a more realistic variables management of the car driver' assistance.

> The codebase produced by the implementation of this project will be available at: https://github.com/omonimus1/ada-spark-car-assistant, from the final deadline of this project, which is set to be 29th of April 2022.