Automating the Verification of OCS in Airbag Control Unit (ACU)

T V Murali Krishna, S Sivaji, Sunil Tej B

Abstract — ***Airbags have been introduced in automobile long back for the safety of the passengers. Though airbags did fairly good in safety of the passengers, its activation was hazardous many a times as it involves in explosives like sodium azide. though it is having deflation mechanism, it even more harmful to the infants and kids.it necessitates to have an automatic occupant's classification system, so as to classify the occupants of the seat. The National Highway Transportation and Safety Administration (NHTSA) has mandated to equip with automatic system to detect the presence of child or infants from 2006 onwards.In this paper we proposed to classify the outcome as four classes namely infant occupant, child, adult of empty so as to take the decisions accordingly. The automation scripts are written in java script. Here the module's functional requirements are scripted into many test cases. So, each time human has to execute each script individually. This process takes more time and also an engineer has to physically sit there. Hence "One Click Automation" is required and Automation framework for ACU modules is designed.***

Keywords—Airbag Control Unit, Occupant Classification Status, Automation framework, Java Script.

I. INTRODUCTION

Passive safety systems are those that reduce the severity of a crash once it had occurred. These include now-ubiquitous technologies such as airbags and seatbelts; make an important contribution to the safety of the passengers inside the car. They act together to keep the passenger safe after the accident.

Airbag Technology has improved safety standards of the automobiles recently in great way. When introduced in the late 80's , most of the vehicles have only one airbag to safeguard the driver from hard steering, that was the reason for driver fatal accidents . As time progresses, technology become more matured, robust and economical and able to find 4 to 16 airbags in one vehicle. The purpose of the airbag in steering is to save the driver when accident happens . Also an airbag is a soft pillow made of nylon that protects passenger's head and chest primarily. The accelerometer is responsible for measuring the severity of the accident [2]. If the crash is very serious the airbag will inflate promptly and subsequently helps in saving the life. Despite the obvious benefits, it poses some risks associated with airbags. Most prominently the speed at which airbags inflate can sometimes cause whiplash, potentially fatal head and chest injuries or concussions.

Spatially at appropriate place a pressure sensor has been placed to detect the weight of the occupant. When passenger occupies the seat, the pressure sensor generates the signals, whose strength is proportional to the occupant's weight and communicated to ECU. The electronic control unit process the data sent by pressure sensor and generates a activation signal, which could be used to ignite the sodium azide for inflation. The strength of the activation signal is dependent on the weight of the occupants. Airbag system is attached with seat belt tensioners for better use. Before sending the activation signal to heating element of the airbag, ECU will interpret about the usage of seatbelt tensioners.

A led indicator on driver dash board conveys that the passenger occupation status. If passenger occupies the seat, pressure sensor can estimate the passenger as one of the predefined classes. Based on the passenger classification dual air bags will activated fully or partially. Generally, fully activated airbags are very dangerous to the passengers.so as to safeguard the passengers, air bags will be having small holes for deflation.

In addition, if ecu detects the moderately strong signal from accelerometer, airbags can be deployed at lesser speeds which indicates, car is involved in minor collision. Weight sensors are used to detect the occupant; however, carmakers are doing research to improve efficiency[1]. Some experimental systems consider the optical images of passenger continuously to determine whether occupant is a child or adult, makes the decision accordingly.

II. VERIFICATION

Verification is essential process for ensuring the preset specifications are being met. In V-Model testing process will go parallel and also called sequential testing team is responsible for drafting testing strategy, sequence and test cases which precisely guarantees to meet specifications. The V-model verification process shown in Fig. 1.

In V-Model Software Development Life Cycle, development and testing activity will be defined based on requirement specification document.

Revised Manuscript Received on February 11, 2019.

T V Murali Krishna, Assistant Professor, Department of ECE, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur, A.P, India 522213 (Email: thokalamurali2010@gmail.com)

S Sivaji, Assistant Professor, Department of ECE, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur, A.P, India 522213

Sunil Tej B, Assistant Professor, Department of ECE, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur, A.P, India 522213

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

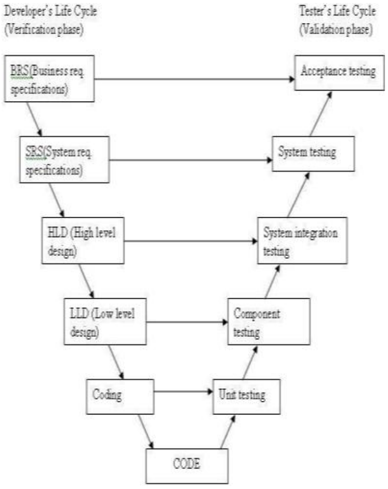


Fig. 1. V-Model Verification

Requirement document will be the basis for developer to propose a design and subsequently testing team will make appropriate test cases as per the strategy. Both activities are working parallel to each other.

The V-model can be termed as Verification and Validation model. The testing activity has been performed in the each phase of Software Testing Life Cycle. In the first half of the Verification model, testing activity was integrated with user requirements and System Design document. In the later half Validation testing will be active. life cycle model generally starts with BRS and SRS similar to waterfall model.in contrast to water fall model, in this model a system test is stared prior to development phase.

The second phase called high-level design (HLD) depends on system architecture and design. It gives the overview of solution, platform, system, product and service/process. After completing the testing pieces of software and integration test was performed to verify the collective working of the software.

Third phase called low-level design (LLD) is meant for design of actual software. In this stage logic is defined for every component of the system. Every component test cases will be prepared in this phase.

Coding: This is at the bottom of the V-Shape odel. Module design is converted into code by developers.

Advantages of V-model:

Simple and easy to use.

Before Coding we have to complete test activities like planning and test designing. This saves a lot of time. Proactive defect tracking will happens ,by using that it is easy to find defects easily at early stage.

It works well for the projects in which requirements are easily understandable.

B. Disadvantages of V-model:

Very rigid and least flexible.

If any changes occurs in the midway then we need to update test document along with required documents.

C. When to use the V-model:

The V-shaped model cand be used for small sizes to medium seized projects where requirements for the test cases are clearly defined.

III. EXISTING METHOD

The existing method for verification of airbag control unit is manual verification. This requires manual interaction. for manual verification no need program.

For testing, the fault is deliberately induced and the system is verified for its correctness. Till now, the testing was done manually which consumes a lot of time and human effort. To overcome this drawback and to be able to test all the features of the product we need to automate the testing process. The test setup for manual testing is shown in Fig. 2.

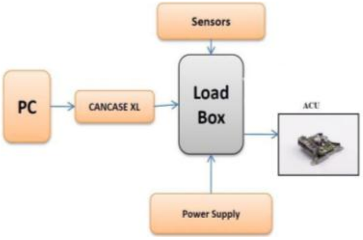


Fig. 2. Block Diagram of Manual Test Bench

The Load Box/Automation Hardware is made with same functionality of the car. Harness is used to connect load box and ACU. It is used for sending and receiving of signals between ACU and Load box. CAN Case XL used for communication between the Load Box and PC.

For the power supply we are using 2 types one is normal power supply of 12V and other is programmable power supply (PPS).Generally normal power supply is used for most of the test cases. For some test cases we may need to test at various voltage levels in such cases we can go for PPS

Crash sensors are located in the front part of vehicle and in the outer body of the car. Each sensor senses the impact in the event of a collision and sends the data to control unit.

Acceleration sensors are to detect the impact of collision. Pressure sensors are to detect physical pressure applied to the vehicle. The status is to be checked for different occupant status like child, adult, etc. Same test scenarios have to run in different voltages levels. Negative test cases are not covered in manual process.

IV. PROPOSED METHODOLOGY & RESULTS

In the automation we are going to use different tools and strategies that reduce the human involvement and Time for Verification.

An Automation framework is a constructive blend of various coding standards, guidelines, concepts, processes, practices, modularity, project hierarchies, test data inputs, reporting mechanism, etc. for automation testing [3]. Thus user can easily understand the guidelines while using the automated test setup scripts. The test setup for manual testing is shown in Fig. 3.

The advantages of automating is in the form of scripting, modularity, scalability, process definition, understandability, cost, re-usability, maintenance etc.

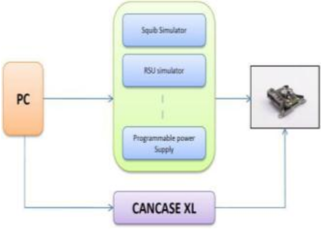


Fig. 3. Block Diagram of Automated Test Bench

Mainly the single standard Test automation Framework arises when bunch of people working on different modules but on the same application and when we want to avoid duplication of the test cases.

Types of testing that can be automated

Functional – testing that operations perform as expected.

Exception or Negative – forcing error conditions in the system.

Regression – testing that the behavior of the system has not changed.

A. Benefits of Automation Framework

Reliable: The tests can be performed precisely when the same operations run several times ,thereby eliminating Human effort

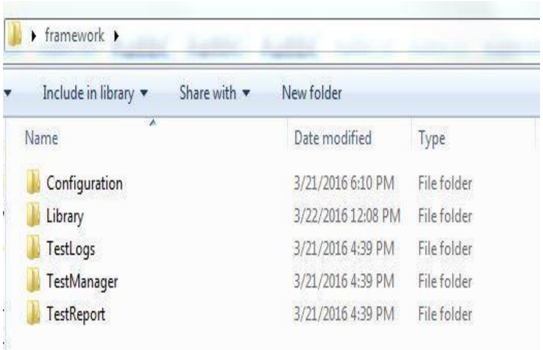
Comprehensive: You can build a test suit that covers possible test case

Reusable: we can use test cases on different versions of an application.

Better Quality Software: we can able to run more number of test cases continuously and within less time with limited resources.

Fast: Automated Tools run tests faster than users.

Cost Reduction: Because of reduction the resources.



Name	Date modified	Type
Configuration	3/21/2016 6:10 PM	File folder
Library	3/22/2016 12:08 PM	File folder
TestLogs	3/21/2016 4:39 PM	File folder
TestManager	3/21/2016 4:39 PM	File folder
TestReport	3/21/2016 4:39 PM	File folder

Fig. 4. Automated Frame Work

In the automation framework shown in Fig. 4, Test manager will run all modules. The flow of automated frame work shown in Fig. 5.For every module a test suit will be created and it is the responsibility of the test suit to call the test cases which are under a specific module. A common li9brary will be used to call all the library functions and configuration files. For every test case particular common library file and configuration files will be called [4].

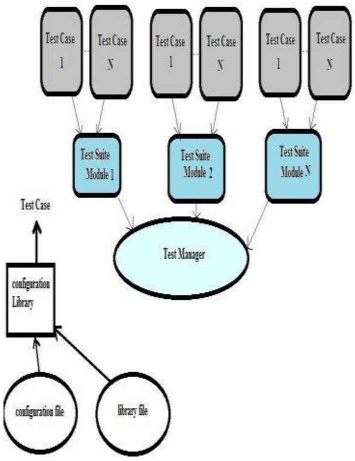


Fig. 5. Automated Frame Work Flow

To check the functionality of a module or system we will be using test cases, for each test case a specific set of conditions will be determined based on which a tester can verify whether all the features of the module are working according to their requirement or not.

A test case consists of number of steps aligned in a particular sequence which are useful for checking the functionality of a application. All the features under a module will be tested and will be compared with the expected result based on the test case requirement.

B. Automated Framework Improvements

Adding all the possible test scenarios which are not presented now in manual

Making Crash is easy in automation. Reusability of automated scripts.

C. OCS Inputs

Supply Voltage – Nominal Voltage, Low Voltage, Large Voltage

Qualification Time – Elapsed, Not Elapsed

De-Qualification Time – Elapsed, Not Elapsed

OCS 1 – Empty ,Small ,Large ,Indeterminate, ALGO cut off defect, ECU defect, sensor defect

OCS3 -- right Vehicle ID, Wrong vehicle id, COMM fault.

D. OCS Outputs

Fault DTC

OCS State – Empty (Class 0), Small (Class 1), Large (Class 2), Indeterminate (Class 3).

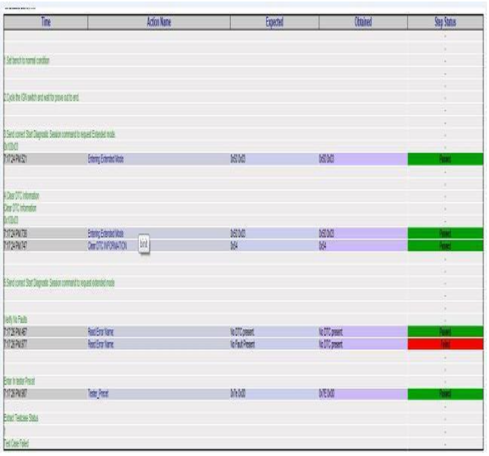
Warning Lamp – On, Off, Flashing, Failure

Telltale Lamp – On, Off, Flashing, Failure

CAN Signal – Default state, Last known state, Valid, Invalid.

E. OCS VERIFICATION FLOW

The verification of OCS starts with OCS3 message Communication fault. If there is no OCS3 communication fault means it will check for vehicle ID (VID). If VID is wrong then Wrong VID fault will be generated then user have to give the correct VID and wait for the wrong VID fault to become historic.The OCS class states and other defects like algo-cutoff defect, ECU defect, Sensor defect are also checked under OSC1 message. The OCS verification flow is shown in the Fig 6.



Time	Action Name	Expected	Obtained	Step Status
1. Set bench to normal condition				
2. Cycle the IGN switch and wait for prove out to end				
3. Send correct Start Diagnostic Session command to request Extended mode				
0x10 0x03				
7:17:24 PM.521	Entering Extended Mode	0x50 0x03	0x50 0x03	Passed
4. Clear DTC information				
Clear DTC Information				
0x10 0x03				
7:17:24 PM.736	Entering Extended Mode	0x50 0x03	0x50 0x03	Passed
7:17:24 PM.747	Clear DTC INFORMATION	0x54	0x54	Passed
5. Send correct Start Diagnostic Session command to request extended mode				
Verify No Faults				
7:17:26 PM.467	Read Error Name	No DTC present.	No DTC present.	Passed
7:17:26 PM.977	Read Error Name	No Fault Present	No DTC present.	Failed
Enter in tester Preset				
7:17:26 PM.987	Tester_Preset	0x7e 0x00	0x7E 0x00	Passed
Extract Testcase Status				
Test Case Failed				

Fig 6. OCS verification flow

The automated Setup software will generate the results automatically after running the scripts. If the step is passed it will show the status as pass in green color as shown in Fig. 7. If the test step failed it will show the step as fail in red color.

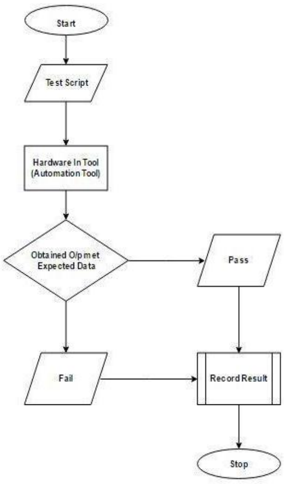

Start
Test Script
Hardware In Tool (Automation Tool)
Obtained O/p met Expected Data
Pass
Fail
Record Result
Stop

Fig. 7. Result

V. CONCLUSION

In this project, automated framework scripting was done for OCS verification in Airbags. This scripts can be reused across the different projects of ACU. The execution of test suite with HIL test setup produces the result of all the test cases under the test suite in a single file. The future work is to produce report file for each test cases available under each test suite instead of a single report file for a test suite and also reducing the difficulty in verifying the test cases.

REFERENCE

[1] M.E. Farmer, A.K. Jain ,“Occupant classification system for automotive airbag suppression”, Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on 18-20 June 2003

[2] Michael E. Farmer , Anil K Jain , “Smart Automotive Airbags: Occupant Classification and Tracking”, IEEE Transactions on Vehicular Technology (Volume:56 , Issue: 1) Jan. 2007.

[3] Vijay D’Silva, Daniel Kroening, and Georg Weissenbacher, “A Survey of Automated Techniques for Formal Software Verification”, Ieee Transactions On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 27, No. 7

[4] Eun Ha Kim, Jong Chae Na, Seok Moon Ryoo, “Implementing an Effective Test Automation Framework”, 2009 33rd Annual IEEE International Computer Software and Applications Conference.