Australian Journal of Basic and Applied Sciences, 10(1) January 2016, Pages: 607-613

AUSTRALIAN JOURNAL OF BASIC AND APPLIED SCIENCES

ISSN:1991-8178 EISSN: 2309-8414
Journal home page: www.ajbasweb.com

Novelty Testing Measures and Defect Management in Automotive Software Development

[1]P.Sivakumar, [2]B.Vinod, [3]R.S.Sandhya Devi and [4]R.Divya

[1,2,4]Department of Electrical & Electronics Engineering, PSG College of Technology, India
[3] Department of Electrical & Electronics Engineering, Kumaraguru College of Technology, India

Address For Correspondence:
P. Sivakumar, Department of Electrical & Electronics Engineering, PSG College of Technology, India
E-mail: psk@eee.psgtech.ac.in

ARTICLE INFO

Article history:
Received 12 February 2016
Accepted 12 March 2016
Available online 20 March 2016

Keywords:
MATLAB, Model Advisor, Model Based Design, Simulink, V-Model

ABSTRACT

Major automotive manufacturers are launching new software aiming for brand survival and to compete in the market. The biggest challenge is to battle to be at the centre of people's digital life taking new dimension every day. Innovative software is designed and renewed every hour towards further improvement and sophistication. In this fast pacing environment a minor defect in the software developed may lead to fatal accidents or malfunctioning of the equipment. Suitable testing measures are vital for the smooth functioning of the vehicle. Currently testing and fault analysis is done only after the Electronic Control Units are developed which makes the finding of specification related defects difficult and costly. This paper solves the problem using model based technique, detecting faults at an earlier stage and remodeling the design economically with less time consumption. Furthermore, Model Advisor is used to check whether the model designed adheres to the functional safety standards such as ISO-26262 and MISRA-C. A Simulink case study model for Adaptive Front Light System is developed and tested.

INTRODUCTION

Road safety means not only adhering to the traffic rules but also the error free design and manufacture of automobiles. Technology in the field of automobiles is moving at an alarming speed towards more perfection and sophistication. Software plays a crucial role in the automotive industry today. Everyone wants to highlight a signature feature of their own. New models are designed with crisp, more expressive styles which have billion lines of code in it. Premium cars feature 70 ECUs connected by five different bus systems. Many automobile industries have recalled cars in thousands due to faulty passenger side airbags (Broy, M., 2006) or problems linked to a remote keyless entry accessory (Ibarra-Alvarado, I., 2005). To avoid casualties, software faults must be identified at the design stage itself. This paper aims to create a model for automotive subsystem and tested at different simulation levels in MATLAB. Model based testing approach is used because it works faster and is economical. Since several steps are repeated, test automation is also required. The high expectation of the public for improvisation in cars put the car industry under pressure. Currently the challenges present in automotive software development are to meet the brand survival demands, competition, quicker time to market demands, increased complexity and functionality, tight performance constraints and high reliability demands (Miller, M.D., R.R. Ulaszek, 2012). The following areas are the main source of innovation in near future – crash avoidance, enhanced driver assistance, advanced technical man-machine interface, driverless cars and networking between cars. As explained above Section I review the necessity for defect management. Section II emphasizes the importance of V-model development, testing methods at various levels and functional safety standards. In Section III, the significance of MATLAB a tool for modeling and testing is discussed. In Section

To Cite This Article: P. Sivakumar, B.Vinod, R.S.Sandhya Devi and R. Divya., Novelty Testing Measures and Defect Management in Automotive Software Development. ***Aust. J. Basic & Appl. Sci.***, *10(1): 607-613, 2016*

IV, study and analysis of the model for Adaptive Front Light System is carried out and the compliance for the functional safety standards like ISO-26262 and MISRA-C is verified. In Section V, futuristic developments are discussed with a broader outlook. Last section concludes with the summary. This paper signifies the necessity of a systematic and comprehensive error treatment in automotive software development.

II. Need for Model based development:

In traditional way of testing verification is done only after the complete implementation of the system. Bugs that occur at an earlier stage can be fixed only at the end. Application area plays a major role in the software development and testing process. Software is usually developed based on agile, lean, waterfall or V-model method. A comparatively better model can be created using model based design. This design starts testing and correcting the system from initial stage of development. Thus the flaws are eradicated with ease. This particular design incorporates Modified V-model based development which includes a branch for safety procedures (Samie, S.A.A.) as shown in Fig.1. Future research work can be carried out with model based testing (Bringmann, E., A. Kramer, 2008) with behavioral model (Conrad, M., 2014).

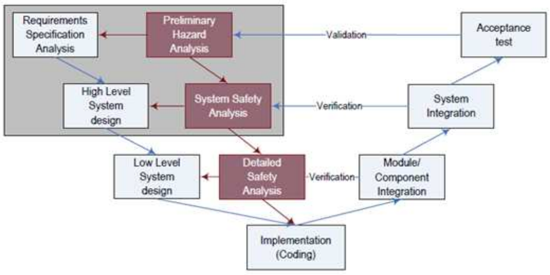

Fig. 1: Modified V-Model Based Development.

The main objective of model based design is to have a proper link between development teams, customers, car manufacturers and suppliers for better communication and understanding. Using models as a perfect reference, requirement planning, design, implementation and testing are linked to reduce complexity (Schoeneburg, R., T. Breitling, 2005). To start with a functional model is developed which can be simulated and then tested. This executable model is used throughout the downstream development process and forms the blueprint for the automatic or manual coding of the embedded software and its integration into the Electronic Control Unit (ECU). Creating models upfront saves more time and by reusing the models, the system engineers are free to focus on design innovation to achieve highly consistent and more efficient performance.

2.1. Testing Levels in Model Based Design:

In embedded automotive software development, testing plays a crucial role. Practically complete testing of the system under study is impossible. Hence the test cases are divided into small number of subsets and testing is done at different levels of integration (Chan, C.K., 2009). Model-in-the-Loop (MiL) is the first integration level. Testing at this level means the embedded system. A model is created and simulated for any physical hardware component and testing at this stage is known as MiL testing. A software is generated from the model developed. Testing the embedded software created is Software-in-the-Loop test and the SiL tests are usually conducted on Windows or Linux OS. ECU hardware is fabricated and tested in Processor-in-the-Loop testing. This test can pinpoint any flaw caused by the target compiler or the processor architecture. The hardware connections of ECUs with I/O services are finalized. The final integration level of testing is Hardware-in-the-Loop testing. Since testing is done at various levels eradication of even a minor error can be done in real-time. Comparatively Modified V-model based development is found to be the best and used in the case study-Adaptive Front Light System. Further work is being carried out with Reactis (Skruch, P., G. Buchala, 2014).

2.2. Automotive Safety Standards:

The model based design should comply with the functional safety standards such as ISO-26262 and MISRA-C.

2.2.1. ISO-26262:

Dependability and reliability of software in modern cars is of utmost importance. ISO-26262 has provided certain guidelines for the progress of both hardware and software to follow certain norms for safety critical applications (Conrad, M., 2010). IEC 61508 is an international standard titled Functional Safety of Electrical/Electronic/Programmable Safety-related system. The standard ISO 26262 is an adaptation of the Functional Safety standard IEC 61508 for Automotive Electric/Electronic Systems (Stürmer, I., 2008). ISO

26262 is a risk-based safety standard, where the risk of hazardous operational situations is qualitatively assessed and safety measures are defined to avoid or control systematic and hardware failures or mitigate their effects. ISO 26262 provides Automotive Safety Integrity Levels (ASILs) which are adapted levels of SIL (Safety Integrity Level) for the automotive domain and are named ASIL A, B, C & D in the increasing order of risk. ASILs are then determined based on the levels of severity, probability and controllability using ISO-26262 standard.

2.2.2. MISRA-C:

MISRA-C is a set of software development guidelines for the C programming language developed by Motor Industry Software Reliability Association. Before the launch of ISO-26262 standard, MISRA-C was used which facilitates code safety, portability and reliability for automotive software. MISRA has two sets of rules – required and advisory where required rules must be satisfied and advisory rules may be optional. In Adaptive Front Light System implementation these above standards are verified and found to be error free.

III. Modeling and Testing Tools:

The study is carried out with MATLAB. Engineers and scientists rely on MATLAB for model-based design. Simulink models are designed applying blocks available in the Library. These Libraries have several types of functions which are used for varied purposes in model development. Stateflow extends Simulink so as to design transition logic. Model Advisor is then used to check the entire model or a sub-system. It is easy to use automated checking during the modeling process. A HTML report can also be generated for the output. Model Advisor is used to check the compliance for the functional safety standards. Any offending block or incomplete models will be located correctly. Simulink is thus found to be an effective tool for the development of Adaptive Front Light System. Apart from Model Advisor, Simulink Design Verifier and SystemTest can also be used.

IV. Adaptive Front Light System Implementation:

As specified in the previous sections a model for automotive sub-system namely front light system is designed using MATLAB and the safety requirements are tested using the Model Advisor and the outcome results are shown.

4.1. Novel Design:

The Adaptive Front Light System control highly depends on the distance of the obstacle. Automatic High/Low Beam System is a system used for automatically adjusting the headlight beams from High beam to Low beam when encountering oncoming vehicles during night time driving, and to switch back to high beam after the vehicles had passed. The effect of glaring due to headlight is also considered.

4.1.1. Automatic High/Low Beam Control with Street Lamp Detector:

A headlamp controlling system can vary the headlamp level and High/Low beam mode according to the distance of the obstacle and the street lamp detector whose model is shown in Fig.2.

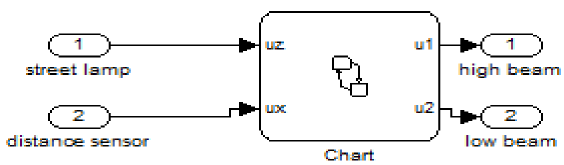

Fig. 2: Automatic High/Low Beam Control with Street Lamp Detector.

The high beam is definitely “OFF” when the street lamp is present since it is not necessary to use High beam in a well illuminated road. When there is no street light, High beam and Low beam will be working according to the distance signal. The light dimming control system will be triggered if the barrier detected is within 5m away. If an obstacle is over 5m and within 50m, the Low beam angle adjustment will be “ON”, and the headlamp angle will be adjusted according to the actual distance. When there are no obstacles detected within 50m high beam and low beam will be at ON state simultaneously. The stateflow for Automatic High/Low beam control with street lamp detector is shown in Fig.3 where transitions between different states are shown.

Australian Journal of Basic and Applied Sciences, 10(1) January 2016, Pages: 607-613

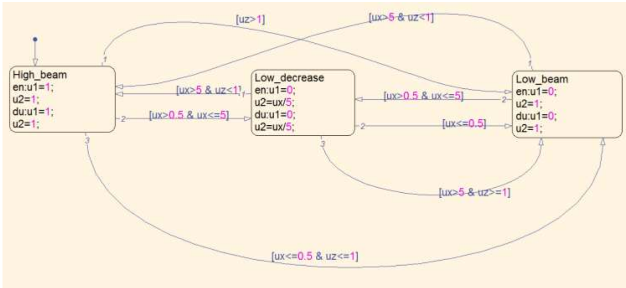

Fig. 3: Stateflow of Automatic High/Low Beam Control with Street Lamp Detector.

4.1.2. Simulation with Front Reflection Sensor:

Assume a light forward sensor has been installed at the dashboard facing towards the front near the level of driver's eye-sight inside the vehicle. The function for that sensor is detecting the headlamp beams reflected by the vehicles at the front causing glare to the driver. The Simulink model for glare avoidance is shown in Fig.4. The High beam is omitted in this case as the testing distance is not over 5m and the High beam is designed to be turned off. If the minimum brightness is over the calculated brightness which the drivers can feel comfortably, Low beam will be turned off completely until the glare reduces. The stateflow for glare avoidance is shown in Fig.5 where transitions between different states are shown.

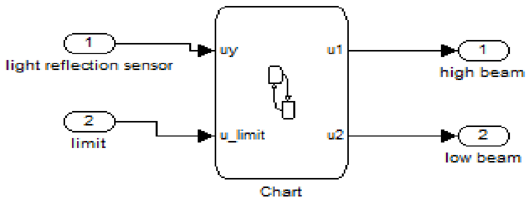

Fig. 4: Glare Avoidance.

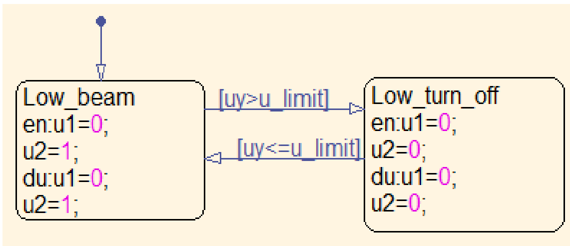

Fig. 5: Stateflow for Glare Avoidance.

4.1.3. Simulation of Adaptive Front Light System:

Gathering all the simulated conditions mentioned above, an Adaptive Front Light system is designed as shown in Fig.6 and its corresponding stateflow diagram is shown in Fig.7.

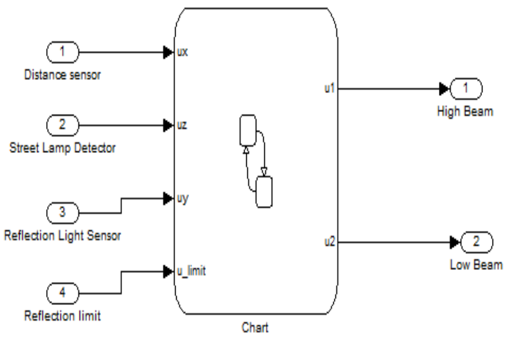

Fig. 6: Adaptive Front Light System.

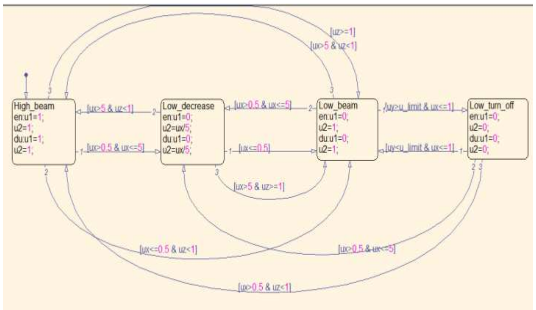

Fig. 7: Stateflow of Adaptive Front Lighting System.

RESULTS AND DISCUSSION

Simulation outcome of three different cases have thus been performed. Model Advisor checks for MISRA-C standards as shown in Fig.8.

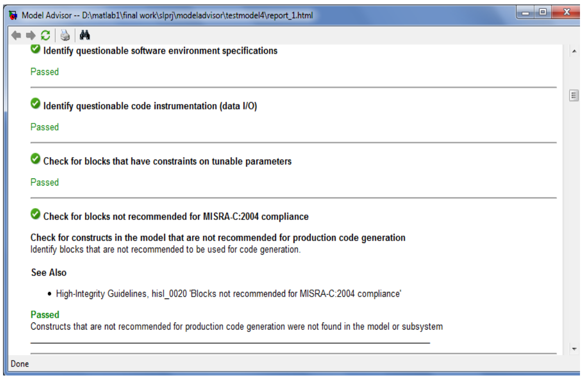

Fig. 8: Model Advisor Report for MISRA-C Compliance.

Model Advisor checks for the requirements for ISO 26262 as shown in Fig.9 which is the Functional Safety standard IEC 61508 for Automotive Electric/Electronic Systems. Thus a model for Adaptive Front Lighting System has been developed and tested with Model Advisor at both model level and code level which is found to be error free.

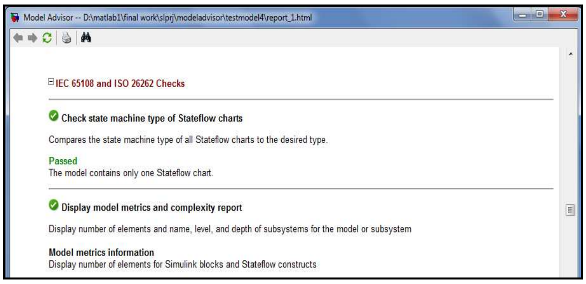

Fig. 9: Model Advisor Report for ISO 26262 Compliance.

As an example to indicate fault present in model developed infinity is introduced in a separate model designed especially for this purpose which will produce dangerous effects in the real-time system. To see where the defect has occurred and how the defect may propagate in the system, Model Advisor is used as shown in Fig.11.

V. Future Work:

This paper mainly concentrates on identifying the faults in the automotive software development process using model based testing. State flow chart helps in the easy development of automotive system models and Model Advisor tool is used for identifying faults at model level and code level. However, there is still an extended way to a fully incorporated and feature rich testing technique for model based testing in the automotive field. One such procedure which is currently being worked is model-based testing using branches, decisions and options which uses finite state machine behavioural model where Test Generator, Test Harness and Test Analyzer are incorporated. Testing tools such as Simulink Design Verifier and SystemTest are to be employed to carry out the above testing procedure. Reactive Systems, a privately held US based company's Reactis product line provides automated testing and validation tools to support the development of embedded control software. It supports model-based design with Simulink, Stateflow, Embedded MATLAB, and C code. Hence this software will be used for further testing.

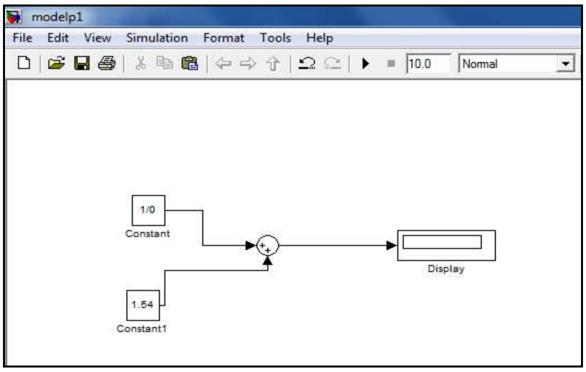

Fig. 10: Introducing Faults in the Simulink Model.

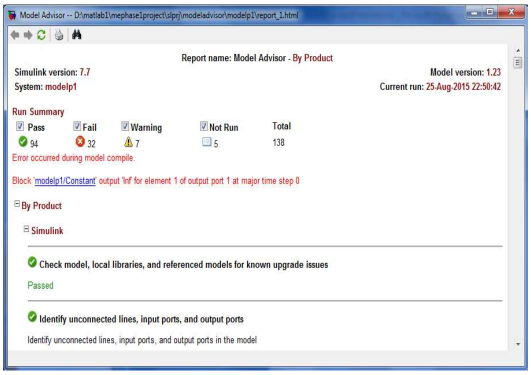

Fig. 11: Fault identified in Model Advisor.

VI. Conclusion:

Model based development under real-time constraint helps to launch new vehicles by reducing the valuable time required for testing. Further study and research works in this field will facilitate automobile manufacturers to develop vehicles that satisfy new government norms that stresses particularly on road safety.

REFERENCES

Broy, M., 2006. Challenges in automotive software engineering. In*Proceedings of the 28th international conference on Software engineering* (pp: 33-42). ACM.

Ibarra-Alvarado, I., R.K. Stobart, R. Lutz, 2005. Software hazard analysis for x-by-wire applications. In *Satellite Events at the MoDELS 2005 Conference* (pp: 341-342). Springer Berlin Heidelberg.

Miller, M.D., R.R. Ulaszek, 2012. *U.S. Patent No. 8,225,288*. Washington, DC: U.S. Patent and Trademark Office.

Samie, S.A.A. Automated Model in the Loop for Embedded Systems Testing.

Bringmann, E., A. Kramer, 2008. Model-based testing of automotive systems. In *Software Testing, Verification, and Validation, 2008 1st International Conference on* (pp: 485-493). IEEE.

Conrad, M., 2014. Verification and Validation According to ISO 26262: A Workflow to Facilitate the Development of High-Integrity Software.*Embedded Real Time Software and Systems (ERTS2).*

Schoeneburg, R., T. Breitling, 2005. Enhancement of active and passive safety by future PRE-SAFE systems. In Proceedings of the 19th ESV Conference, Washington, DC, USA.

Australian Journal of Basic and Applied Sciences, 10(1) January 2016, Pages: 607-613

Chan, C.K., K.W.E. Cheng, S.L. Ho, T.M. Fung, 2009. Simulation of the control method for the adaptive front lighting system. InPower Electronics Systems and Applications, PESA. 3rd International Conference on (pp: 1-4). IEEE.

Skruch, P., G. Buchala, 2014. Model-Based Real-Time Testing of Embedded Automotive Systems. *SAE Int. J. Passeng. Cars–Electron. Electr. Syst*, *7*(2).

Conrad, M., P. Munier, F. Rauch, 2010. Qualifying Software Tools According to ISO 26262. In *MBEES* (pp: 117-128).

Stürmer, I., C. Dziobek, H. Pohlheim, 2008. Modeling Guidelines and Model Analysis Tools in Embedded Automotive Software Development. In *MBEES* (pp: 28-39).