Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 78 (2016) 851 – 858
Procedia Computer Science

International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA

New Approach for Testing and providing Security Mechanism for Embedded Systems

Swapnili P. Karmore[a], Anjali R. Mahajan[b]

Research Scholar, Department of Computer Science and enginnering, G. H. Raisoni College of Engineering, Nagpur, India.
Head, Department of Information Technology, Government Polytechnic, Nagpur, India.

Abstract

Assuring the quality of embedded system is posing a big challenge for software testers around the globe. Testing of one embedded system widely differs from another. The approach presented in the paper is used for the testing of safety critical feature of embedded system. The input and outputs are trained validated and tested via ANN. Security is provided via skipping invalid and critical classes and by embedding secret key in the Ram of embedded device. The work contributes simulating environment where cost and time required for testing of embedded systems will be minimized, which removes drawback of traditional approaches.

Keywords: Embedded system testing; Black box testing; Safety critical embedded systems; Artificial neural network.

1. Introduction

Testing is the most common process used to determine the quality and providing security for embedded systems. In the embedded world, testing is an immense challenge. In the test plan, the distinguished characteristics of embedded systems must be reflected as these are application specific. They give embedded system exclusive and distinct flavor. Real-time systems have to meet the challenge of assuring the correct implementation of an application not only dependent upon its logical accuracy, but also its ability to meet the constraints of timing. To provide security and to implement real-time embedded system must be predictable because of the need to meet the timing requirements. Thus, this must be taken into account while their supporting software is being written.

* Swapnili P Karmore. Tel.:+91 7875474053;
E-mail address: swapnili.karmore@raisoni.net

Peer-review under responsibility of organizing committee of the ICISP2015
doi:10.1016/j.procs.2016.02.073

Each embedded system is an application precise system and requires a separate tool or separate code for testing. Presently, there is no such mechanism available in the market, which is competent enough to check a number of embedded systems under one platform. Testing is the most time consuming and costly portion of the embedded system development process. The quantitative and qualitative analysis from an industrial survey says that continuous efforts are required in the testing process to ensure improved quality and efficiency in the development of embedded systems.

1.1. Types of Testing

In the case of embedded systems, testing can be performed for different reasons, sometimes it may be for testing of the inputs and sometimes for monitoring software state and outputs for expected properties. Examples of these are like matching up to the expected output values which can then be used to check whether the product covers all the requirements specified before or to save the systems from failure. Verification and validation are the important activities in the testing process. Verification is the process of human examination or review of the work product. Validations are required when there is need to execute the things on computers. There are several styles for testing and depending on various situations several types of testing can be conducted on the embedded systems.

The white box testing, the black box testing and the exploratory testing are the testing techniques, which are used as per the requirement of the system under test. The various types of testing conducted throughout the life cycle are unit testing, subsystem testing, integration testing, regression testing, acceptance testing and beta testing. An exploratory testing type is performed by human observers. They have to exercise the system as well as search for unexpected results. Exploratory testing is designed to find those strange behaviors, which were neither defined in requirements nor in the specifications. The defects, missed in the standard testing methodologies can be traced in Expository testing. The disadvantage of this testing is the lack of document measurement coverage. While performing this test many defects remain uncovered.

Black box testing is usually referred to for embedded system testing. It is designed after considering knowledge of behavior, when it is designed without considering knowledge it is known as functional testing. Black box testing is usually associated with functional behavior without knowing inner details. The advantages associated with black box testing are that these can be used for testing the final behavior of the system and can be written independently of software design.

White box testing is testing of internal structure of the program and it is also called the structural testing. In this type of testing, control flow, condition branches and structures are exercised. The advantage of white box testing is that it gives good coverage of tests and tests are basically designed for maximum coverage. Boundary and special cases are tested successfully in this type of test. Test operation covers every point in the lookup table and in the control loop; it also exercises both paths of every conditional branch statement. Disadvantage of this technique is the number of functions remains uncovered.

The possible types of testing conducted in embedded system testing are usability, boundary conditions, performance, state transitions, load testing, main stream usage against a scenarios security and error recovery. Compatibility, configuration, equipment variations; old versions, instability and serviceability can also be checked in embedded system testing. Regions where programmers make mistakes are user interfaces and boundaries. The errors occur because of incorrect error handling, missed error checks, race conditions, load conditions and calculation errors because of application of wrong algorithms

1.2. Selection of Right Approach

Selecting the right approach is a very important task in the testing of embedded system. It is a very big question as to how can we decide that a specific approach is appropriate for a specific embedded system. Some sets of evaluation criteria can be used for selecting the appropriate approach, e.g. to what extent the organizational structure can provide the assistance for rapid decision making, whether it encourages teamwork or provides an independent, unbiased, formal, strong and rewarded, test organization. For selecting the correct approach we should map the current organization, define and prioritize the key selection criteria & document the new potential approaches, using a selection grid. Decision matrix selection grid is one of the quality tools, which can be used to evaluate the various approaches.

After selecting criteria they are numbered or uniquely identified and placed across the top of the decision matrix. Table 1 shows the diagrammatic explanation of the tool. In this matrix various approaches are vertically placed and numbered and criteria are listed as 1 to 5 where 5 indicates higher ranking 3 is for medium ranking and 1 is for low ranking. After selecting proper testing approach embedded system can be easily tested. Testing of embedded system may be based on software or hardware of embedded system. Sometimes it is used to test individual component of embedded system and some time it is used for the functioning of modules present in embedded system. The testing of embedded system based on its infrastructure, its life cycle, techniques used and its organization.

Table 1. Decision Matrix Grids.

Approach/Rank	1	2	3	4	5	6	Total
Approach 1	×						
Approach 2		×					
Approach 3			×				
Approach 4		×					
Approach 5				×			
Approach 6						×	

The first section contains a detailed introduction of the work and its importance. Section two involves review of existing methodologies related to the embedded system testing. The conclusion of literature survey of section two leads the need for new approach. In section three, the proposed plan is presented. It discusses the block diagram of new approach. Section four outlines the results obtained while obtaining a new approach for testing of embedded systems. Section five outlines the conclusion and the future scope.

2. Literature Survey

In this section, the existing methodologies of the embedded system testing are discussed. Embedded system testing is a very complex, costly, critical and time consuming process. Researchers have used different tools and platforms for testing of the embedded systems, but no one has tried to provide a common tool, where all the embedded systems can be tested together. Temb method is the traditional method used for the testing of embedded systems. It provides a mechanism to a dedicated test approach for assembling, which can be applied to any test project. Temb method consists of test project planning, techniques, test environments, test team organization and reporting processes. The mechanism which is used for the assembling of test approaches is based on the examination of the risks and characteristics of the system. Temb method assembles generic and the specific measures. Embedded system has two main characteristics that are safety critical and technical scientific, which focus on covering a system diversity using limited set. It gives information about what is needed in the dedicated test approach and is made concrete by analyzing relevant characteristics. In Temb generic elements constitute the structured testing process. The four cornerstones help embedded system ensure that vital things are not forgotten and make the testing approach more controllable and manageable. As Life cycle helps to show more activity than the test organization, it distinguishes between things arranged more properly and effectively[1].

Researcher proposes MDT of real time embedded system and orientation of object oriented model to the function oriented model development. Now a day's Automation is essential in the testing of embedded system testing field.

Test automation using object oriented approach is the ultimate goal of the researcher. Because of the diversity between the languages used by developers and the testers there exists a big communication gap[2]. Standard system requirements and the methodologies need a bridge to remove this development gap. UML and Simulink comparison was performed in the proposed literature by the researcher in order to develop an efficient mechanism, the methodology used was model driven. In this literature, methodology used is object oriented and automation. Development processes were also implemented simultaneously[3]. Signals have been given as inputs to the model, which further lead to system and test implementation section.

Another existing approach proposed by the researcher suggests MLCH designs for embedded electronics systems. Heterogeneous systems were taken under consideration and model level coupling was performed through four layered Meta model architecture. Rapid prototyping approach was used for the testing of embedded systems. The Number of ECU used in the automated car was the objective behind the research here. Integration was performed via UML and the tool was able to support all the programming languages. Code generation and integration was performed with Matlab. Techniques used for testing of embedded system were white box and the block box[4].

The other approach was rapid embedded system testing, where verification pattern were used. For reducing time required for debugging and efforts, reuse of the proven test scripts was conducted. Verification framework was developed on the basis of which detailed architecture was drawn. Event driven verification and logic was the key component in the approach. Verification process included collection, matching and generation of test scripts as well as executing the test scripts with verifying results[5]. Messy GA has introduced and suggested in another piece of literature to automate the test cases with State flow model testing where the search based testing was performed via simulink/state flow models. The approach used was SBST. The test data generation algorithm with the experimental setup and the TC results were discussed in the overall research[6]. Black box testing approach is generally used in the embedded systems. The research concentrates on the software developed for the development of the embedded system. Software model, referred in the research work uses MATLAB for conducting tests, also known as model in the loop designed for embedded system test[7]. Neural network is used for the testing of embedded systems[8].

3. Proposed Methodology

This work considers testing in several phases, including identification, categorization, static analysis, model checking, pattern generation and training. Major points of focus are safety critical characters of embedded system which make them real time embedded system[16]. The black box technique is used to test the functionality of an embedded system. Proposed methodology gives a new method of testing real and virtual real-time embedded system. The concept of classification is used to solve the problem of testing of embedded system. The input and outputs are trained, validate and tested via ANN, to find out the valid and invalid classes, to improve the system accuracy and to minimize the failure rate of embedded system. The test procedure consists of designing and development of various embedded system. An embedded system requires a number of sensors. The sensors are responsible to demonstrate the safety critical behaviour of the embedded system. Misbehaviour of input mechanism can lead failure of output mechanisms. ANN is responsible to skip invalid and to manage the critical section for providing security to the embedded system.

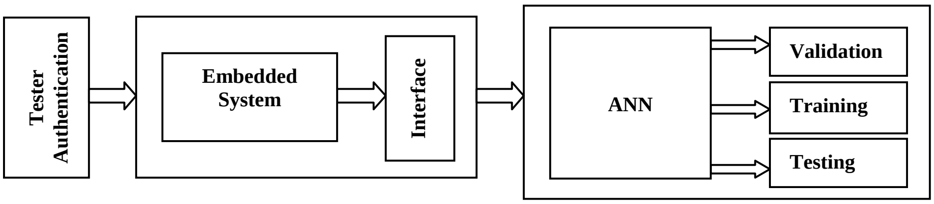

Fig.1. Embedded System Testing Block Diagram.

Maintaining confidentiality in the development of embedded system is a need for service providers. The tester authentication module is developed in order to conduct secure testing mechanism. The secret key is embedded in the

ROM of embedded device in order to provide security to access embedded data as well as in the interface. Mostly embedded system is concerned with medical devices where data is sensitive or it is used in defence, where again security of data is an important constraint. Testing process is secured via tester authentication mechanism.

3.1. Testing of Safety critical function of embedded system.

Embedded system used, is named as car crash avoidance using safety bumper. In this system, the speed of the car is controlled via detecting the force applied on a bumper. Since force is the function of velocity and displacement, input is considered as displacement, while velocity output function. If speed of a car is out of control, critical condition may occur. A car having a mass m of 1500 Kg hits the bumper at the speed of 90 Km/hr. controlling the speed of the car takes place within the range 0 to 3 m range ($0 \leq x \leq 3$). Speed of the car is directly proportional to force exerted on bumper, when it increased more than 100 above the force experienced on the bumper increases.

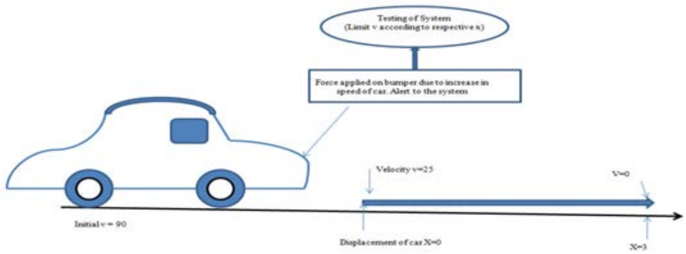

Fig. 2. Embedded System Safety critical function concept Diagram.

A sensor attached to bumper alerts the driver to provide control on the car. The range of inputs is varies from $0 \leq x \leq 3$ and range of output is varied from $0 \leq v \leq 25$. The mass of the car may vary from system to system. The applied force on the Bumper is zero, when the speed of the car is 90Km/hr. The mentioned state is defined as a zero displacement. The bumper attached to the car is designed in such a way that the force experienced on the bumper in out of control case of car, is a because of the velocity v and the displacement x of the front end of bumper according the equation. Where, $K=30s\text{-}kg/m^5$ is constant.

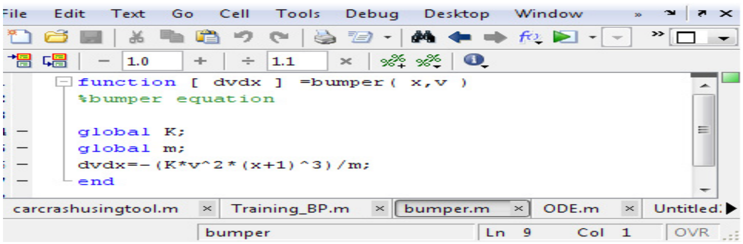

Fig. 3. Embedded system function equation.

The classification concept of AI is used in the work, Classification exists after checking input data i.e. displacement and output data i.e. velocity to define the state of car. The state is defined as follows:

- If (25<v || 25>v) && x>3, state is Critical state.
- If (v<0 && x<3) || (v>0 && x<0), state is Invalid state.
- If (0< v<25) && (0<x<3), state is Valid state.

Where, x is the displacement parameter and v is the velocity parameter. The Figure 3 shows the programming for function used in the embedded system, which is responsible to make it safety critical embedded system. Here displacement and velocity values will responsible to calculate the force exerted on the bumper. If these values will not be properly calculated the car controlling mechanism will not able to detect the speed of the car.

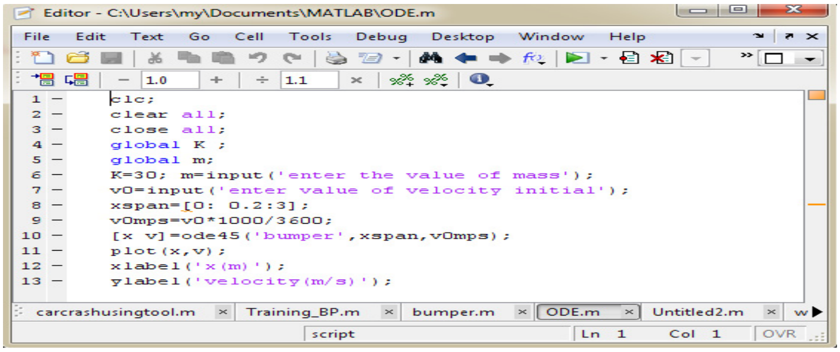

Fig. 4. Snapshot Program for Speed Detection System to Avoid Car Crash.

The table number 2 gives the design of test case to describe the safety critical function of the embedded system. The first portion of test case describes the test information, including test case ID, test case description, preconditions and the second portion of the table gives steps performed in testing, step description, test data and status. The pass or fail status will recorded and fail status will skipped to avoid the failure of embedded system.

Table 2 Test case design for speed detection function of embedded system

Test Information	Test Functions		
Test case ID:	Fun_1(Function ID)		
Test case description	Speed detection system for car crash avoidance(Function description)		
Preconditions	Mass =1500 Kg, Speed=90km/h at x =0(Functions parameters values)		
Step No.	Step description	Test data	Status
1	Calculation (velocity at particular displacement)	v.mat, x.mat	Pass/Fail

In the automated vehicle number of safety critical functions can be used for providing security to the vehicle. These safety critical functions are further attached to sensors in order to indicate that whether the system is running well or not.

4. Results

In this section relevant results are outlined to authenticate the framework and to justify the objective. The classes are developed in order to find the valid and invalid input states. The figure 5 shows the bumper equation graph.

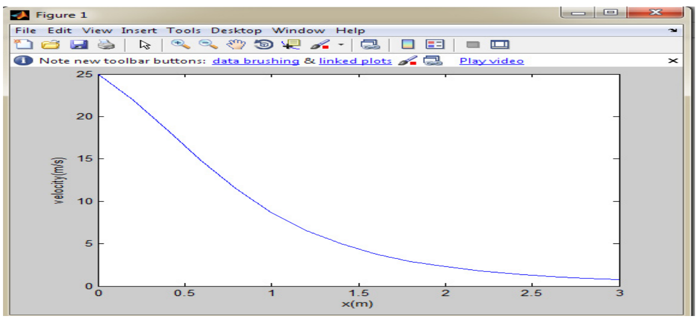

Fig. 5. Nonlinear system graph velocity verses displacement.

The results mention the critical class, where the embedded system can be proved as safety critical embedded system. This section gives results and conclusion obtained in experiments performed in the testing process of embedded systems. The Critical classes are managed and invalid classes are skipped in order to avoid failure of embedded system.

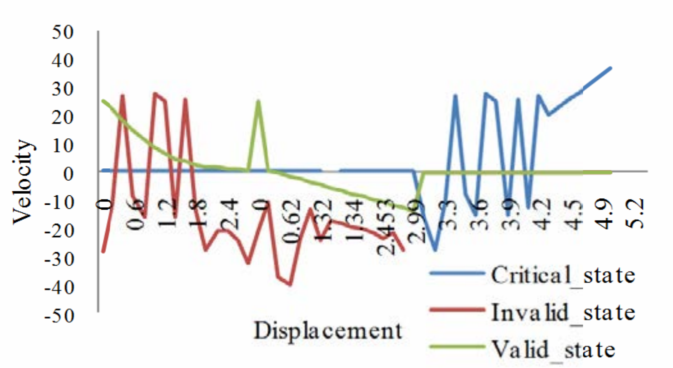

Fig. 6. Graph Classification of States of Speed Detection.

The main program calls this function to test the embedded system. The figure 6 shows the classification of the critical, valid and invalid class's distribution obtained after performing experiments. Table number 3 shows the observations related to pass and fail states.

Table 3.Observation Table

Classified state	Training Error rate (%)	Testing Error rate (%)	System's Test Observations
Critical state	87 88 80 89 87.88	84.4 83.245 79 90 87	Fails at x=3, Test is valid for x>3, critical states are identified and managed.
Invalid state	90 91.12 92.13 89 87	89.99 90.2 91.23 88.23 86.78	Test is passed for invalid state when velocity is negative in the range of 0 to 3 (displacement).
Valid state	89.45 90.23 99 98 92.34	88.88 89.9 98.678 97.46 90.24	Test is passed for its valid state where velocity is positive showing good Performance value.

5. Conclusion

The sensor activity is responsible to decide failure or success of embedded system. One electronic control unit is responsible to test multiple functions. The input and output values are monitored tested and validated on the basis of some fixed rules. The ANN is used to perform validation training and testing of input and output data. New approach uses a classification method of artificial Intelligence, which is responsible to generate three classes of conditions. Valid, invalid and critical regions are categorized on the basis of the classes, the tester can easily find out the misbehavior of the embedded system arises because of input and outputs. The neural network fitting tool uses LMA for training, validation and testing. The work is performed in order to provide security to embedded system by identifying the risk regions and it avoids the use of real embedded systems in experimental work as cost of embedded system is not affordable to the tester. It will prove to be the most promising and innovative approach, if it would be made open ended for the further researches. The framework employed different techniques in order to make the approach generalized and efficient. Empirical evaluation of the framework was performed using a variety of embedded devices under test. The relevant evaluation showed that the framework will able to test the number of embedded systems together under one roof in the future. This new approach is responsible to provide security to embedded system by providing an efficient testing mechanism. The work contributes simulating environment where cost, time and efforts required for testing of embedded systems will be minimized in order to remove the drawbacks of traditional approaches. The classification concept is used for testing of nonlinear embedded system.

Acknowledgements

The authors would like to thank Dr. Preeti Bajaj, Director G.H. Raisoni College of Engineering and Dr. L. G. Malik, Head of Department of Computer Science and Engineering for their constant support and motivation.

References

1. Ayaz Farooq and Farooq, Ayaz. An Evaluation Framework for Software Test Processes. ULB Halle, Germany, HALCoRe; 2009.
2. A Zundorf, L.Geinger, J.Siedhof. Model-Based Development of Embedded Systems. 10.-14.012005, *Technischer Bericht*, TUBS-SSE-2005-01.
3. Justna Zander-Nowicka, Zhen Ru Dai, Ina Schieferdecker. Model Driven Testing of Real-Time Embedded Systems-From Object Oriented towards Function Oriented Development. Berlin Germany; 2010.
4. Clemens Reichmann, Marcuhs Kuhl, Philip Graf. General Store-A Case -Tool Integration Platform Enabling Model Level Coupling of Heterogeneous Designs for Embedded Electronic Systems. *11th IEEE conference* and Workshop; 2004.
5. Wei-Tek Tsai and Lian Yu, Feng Zue.Rapid Embedded Systems Testing Using Verification Patterns. *IEEE Computer Society*; 2005.
6. Jungsup Oh, Mark Harman, Shin Yoo. Transition Coverage Testing for Simulink / State Flow Models using Messy Genetic Algorithms. ACM, Dublin, Ireland; 2011.
7. Justyna Zander-Nowicka. Model-Based Testing of Real-Time Embedded Systems in the Automotive Domain. Berlin; 2009.
8. Ruilian Zhao, Shanshan Lv. Neural-Network Based Test Cases Generation Using Genetic Algorithm. *2013 IEEE 19th Pacific Rim International Symposium on Dependable Computing*, pp. 97-100.
9. M. L. Bushnell and V.D Agarwal. Essentials of Electronic Testing. *Kluwer Academic Publishers*, Norwell, Ma; 2000.
10. Causerie, A. Sch. of Innovation, Design & Eng. An Industrial Survey on Contemporary Aspects of Software Testing. Software Testing, Verification and Validation. *(ICST)*; 2010.
11. Swapnili Karmore, Anjali Mahajan. Development of Novel Approach for Testing of Differential Embedded System. *6th International Conference on emerging Trends in Engineering & Technology ICETET2013*, IEEE Conference 16-18 December; 2013.
12. Swapnili Karmore, Anjali Mahajan. Universal methodology for testing of embedded system. ICCSE2013 International Conference on Computer Science and education 2013, *IEEE Conference*; 26 to 28 April 2013 Colombo, Sri Lanka.
13. Swapnili Karmore, Anjali Mahajan. Testing of Various Embedded System with Artificial Intelligence Approach. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 3; March 2015.
14. Swapnili Karmore, Anjali Mahajan. Test Algorithms for Embedded Systems Testing. *International Journal of Scientific and Engineering Research*, Volume 6: Issue7; July 2015, ISSSN 2229-5518.
15. Swapnili Karmore, Anjali Mahajan. Testing of Embedded System, An Issues and Challenges. *International Journal of Enhanced research in Science, Technology & Engineering* ISSN: 2319-7463, Vol. 4, Issue8; August-2015.
16. Jonathan Bowen. The Ethics of safety critical system. April 2000/Vol. 43, No. 4, *Communication of ACM*.
17. M.I.A.Lourakis. A brief description of Levenberg-Marquardt algorithm implemented by Levmar. *Technical report .Institute of Computer Science, Foundation for Research and Technology* –Hellas; 2005.
18. Jang-Jin Kwon, Dept. of Comput. Sci., Chungbuk Nat. Univ .Collision Analysis of Safety Devices to Prevent Hazards in Safety Critical Systems. Software Security and Reliability (SERE). *2014 Eighth International Conference on June 30 2014-July 2 2014, San Francisco*.
19. Srivaths Ravi and Anand Raghunathan. Security in Embedded Systems: Design Challenges. *ACM Transactions on Embedded Computing Systems*, Vol. 3, No. 3; August 2004, Pages 461–491.
20. IEEE. Ieee standard testability method for embedded core-based integrated circuits. *In IEEE Std* 1500-2005; 2005.
21. Amitabh Das, Dusko Karaklajic and Ingrid Verbauwhede. Secure Mutual Testing Strategy for Cryptographic SoCs.