Development of Application Programming Interface for Testing the Services of Embedded System

Goldi C. Jarbais
Department of Computer Sci. &Engg.
G. H. Raisoni College of Engineering, Nagpur.

S.P. Karmore
Department of Computer Sci. &Engg.
G. H. Raisoni College of Engineering ,Nagpur .

Anjali Mahajan, Ph.D.
Department of Computer Sci. &Engg.
Priyadarshini Institute of Engineering and Technology,Nagpur.

ABSTRACT

Target based testing ina embedded system is a complex and difficult task. This paper, in the perspective of software solution, presents a generic API for testing the target embedded system. Companies developing products based on embedded systems understand the fact high quality and unique application software plays an important role in the differentiation of their products from their competitors.This generic Interface can test the various services of the embedded system, and supports unit test and integrationtest which are based system testing technology. By using generic interface in the target embedded platform to do testing the results show that our constructed testing interface is workable.

General Terms

Development of API,testing of embedded system.

Keywords

TBES:Target based embedded system,middleware software,unit testing,integration testing.

1. INTRODUCTION

Testing is a process centered around the goal of finding defects in a system. Debugging reasons or acceptance reasons are considered – trying to find defects is an essential part of every test process. Testing is an essential element in system development – it helps to improve the quality of the system. Testing is a part of the total quality management within the system development process.

Embedded systems are widely used in various technical domains and applications. Recent embedded systems use quite complex microntrollerscomprising processors, memories and various peripheral circuitries. Hence testing them is a big challenge. In the case of embedded systems dedicated for specific applications hardware resources are used in a limited range moreover according to some fixed scenarios. This creates the possibility of simplifying and increasing effectiveness of the testing process.The method used called asa application driven testing.Testing is more than exercising the system and also involves the correctly behaviour of the system. It also includes the, planning of activities, test cases are designed, managing the test infrastructure, organisation setting , handling the politics and much more.We used the non-volatile memory to stored the embedded software.The volatile memory may be ROM but it can be hard disk.or flash card and downloaded. The embedded software is compiled for a particular target processor, the processingunit,which also required the RAM to operate.The processing unit handles all input and output (I/O) of signals through a dedicated I/O layer.

2. LITERATURE SURVEY

Testing of any embedded system will be significantly different from testing of other embedded system for example the testing of mobile phone is different from the testing of video set top boxes or cruise control in cars.

For each characteristic specific measures (solutions) exist that apply to the specific situation and can be included in the test approach. Through this mechanism the suitable dedicated test approach can be assembled from "generic elements of any test project" and a set of relevant "specific solutions" related to the observed system characteristics of the embedded system [1].

Although many reasons exist why different embedded systems must be tested quite differently, there are also many similar problems and similar solutions that are part of any test approach. For instance: according to certain life cycle,standardized techniques are applied for the planning of test project, particular test environments, organizing test teams, formal reporting, etc.They are the generic elements, covering the four cornerstones of structured testing: Lifecycle; Infrastructure; Techniques; Organization.

All the four cornerstones must be covered equally.Lifecycle can be considered as the central cornerstone.Lifecycle holds the other three cornerstones and at different stages of lifecycle other will be applied.If any one of the cornerstone is avoided then the test process will run into trouble.

Though all the important aspects are considered but it is not guaranteed that the test process will never run into

trouble.But with the help of structure test process.there are chances to recover from such events with minimum damage.

It can be considered the basis of any test approach. For a particular embedded system, this basis test approach must be used with several specific measures to handle the specific problems of testing the particular system.It can be called as the specific solution layer.Specific test design techniques can be applied to test the state based behaviour of the Model is developed in real time environment for testing the system.

Each embedded system software have their own characteristics, hence, the test interface design methods and the test script statements of different modules belonging to the different test systems, cannot be exactly the same. However, for a fixed unit, it is entirely possible to design a standard module test interfaces and test script templates that are suitable for the unit based on the characteristics of the products and the development environment of it. For the different embedded systems, you need only to do a little work to transplant them into a new system, and then achieve the module test automation and formalization of the new systems and then improve the test efficiency. Companies developing products based on embedded systems understand the fact high quality and unique application software plays an important role in the differentiation of their products from their competitors.

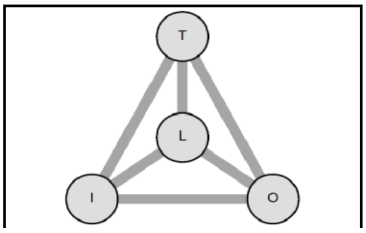

Figure 1.The four cornerstones of a structured test process

Simulation based verification [2] is used in order to test the multicore applications.The automated test generation framework is created with the static analysis techniques.

Multicore communication API(MCAPI) standard is emerged for the multicore embedded system.It involved the mutation testing and model checking.The MCAPI used the message passing programming library for the embedded system.The paper focused on developing the techniques to improved the testing of multicore software.

The MCAPI standard used the previously generated mutation library in order to target message passing communication constructs. The counter example produced by the model checker served as test case in order to determine whether the original program and the program obtained by inserting a fault are equivalent or not.

Preliminary experiment are limited by the capacity of the model checker.Again due to the lack of space the connectionless messanger are used in order to implememt the communication.MCAPI used the C bounded model checker(CBMC) which removed the implementation of shared memory operation.It is found that the MCAPI lacks the larger benchmark because of which the effectiveness of the techniques cannot get explored.

The method [3] used LPC1766 as the micro-controller to build a USB download interface circuit to download the data of POS machine to the external storage.It helps the user in case, when network get breakdown or interrupted.

With the help of interface the user can access the data from the POS machine as per there convenience.As the POS machine is connected to the network it also solve the reliability problem to some extent.

In order to design the interface Cortex-M3 Microcontrollers of LPC1766 is used as a main controller chip.The internal ATX of USB device controllers the transmit/receive bi-directional signal D^+ and D^- of USB bus.

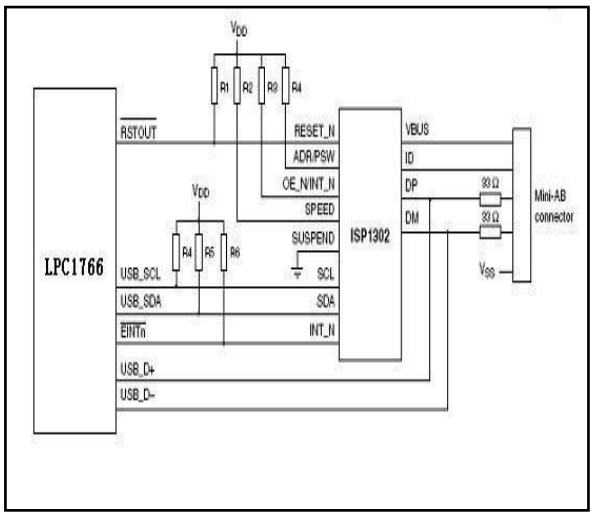

Figure 2.LPC1766 USB OTG port configurations

The diagram shows the USB interface hardware circuit with ISP1302 chip.It can receive or transmit the serial data at the highest speed of 12Mbits/s or at the lowest speed of 1.5Mbits/s.

In addition to the network transmission model a USB data download interface were also presents in the Embedded POS. One of the limitation related to the implementation is that when the system goes into the USB connection mode all the other executing task has to be closed, in order to make sure of high speed communication between USB and master device.USB download interface made it possible to download the data from the internal register to the external register,so that the user can preserved and managed the data.

As the embedded systems are diverse in architecture it caused the problems related with the used of code,portability and dependability.With the help of middleware [4] the problem can be solved to some extent.It help to provide the interface which can be scalable, transparent. Middleware is a software between operating system and an application to solve many beginingproblems.The application of touch screen for the embedded middleware system is design in order to verify feasibility.The functionality and integrated test of embedded middleware system can be access by S3C2410 and XScalePXA270 system.Communication between system to system ,operating system to libraries ,libraries to application is the issues related to the diverse architecture of embedded systems such issued were solved to some extent.

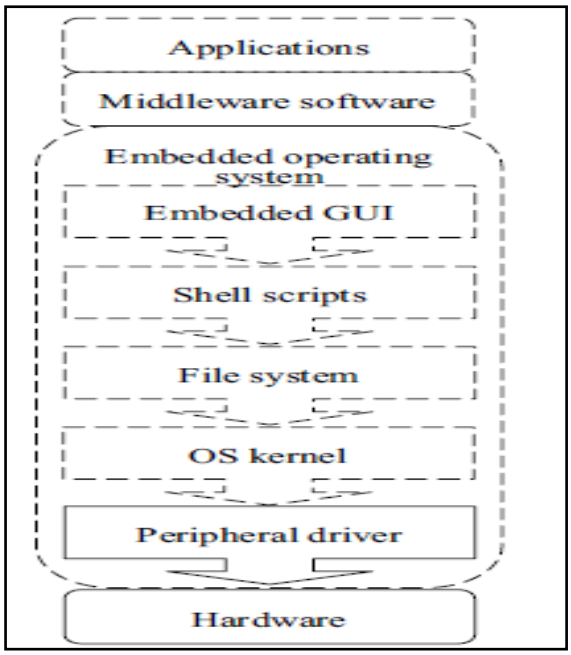

Figure 3.Architecture of embedded systemwith middleware software

The figure shows the architecture of embedded system with middleware software.It consists of four modules.From top-down the modules are application, middleware software,embedded operating system as well as hardware.

The middleware consists of API module, service manager and content manager module.API module is used to solved the problem related to portability and dependability.API module is used with interface for communication with lower and upper layer.The service manager is used for generation and deployment content for requirenment of application.The module can also provide the related content for various application.The module is design for achieving the flexibility and scalability for embedded system.The other module,content manager is used for conducting the content information.It is used to solved the transparent issue.

The algorithm for SAT(SATisfiability test for propositional formula) is not applicable directly for the real-time diagnosis due to its computational complexity.In 2008,Satoshi Hiratsuka et al,[5] transform the constraints in Sign Algebra into the set of the Boolean equations. The diagnosis algorithm is described in the form of logical functions that are parameterized solutions of the equations, and implement these solutions in logical circuits in order to achieve the more accurate and quick responsible diagnoses.

Ahyoung Sung et al, [6] discussed a test data selection technique using a fault injection technique for interaction between hardware and software.Requirement specification are used in order to simulate the behavior of embedded system to software program.The hardware fault are converted to software fault in order to select effective test data selection technique.As the embedded system is a combination of hardware and software,so it is supported to have unexpected failure in interaction between hardware and software. Therefore it is essential to develop a test data selection technique.Test data selection technique for testing the embedded system consists of two steps.First step is to analyze the software and hardware module and second step is to select test data for testing embedded system as shown in figure.

From figure the block HBD represents the hardware block diagram and the block HSBD represents the hardware software block diagram. As shown in Ficture 1, the program S simulates behaviors of the HSBD. The software faults converted from hardware faults are injected into the program *S* to compose the program *P*s.

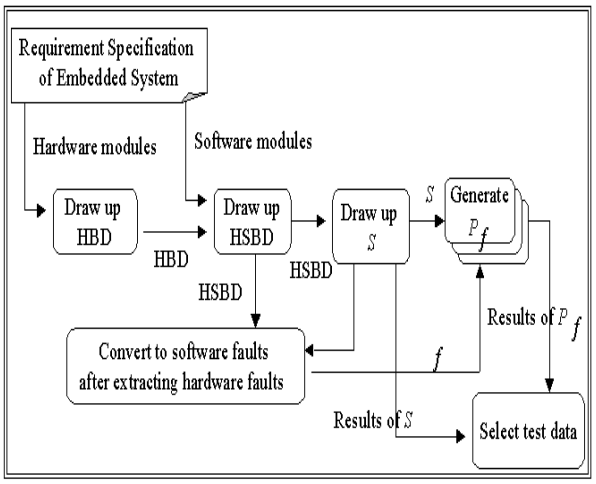

Figure 4. Procedure of Embedded System Test

In 2009,Hua-ming Qian et al, [7] discussed the embedded software testing process model is put forward with the shortcomings of software testing V-model.Again it is observed that the performance of the embedded system is affected by the various architecture and the design.Target-based testing: Based on the software requirement specification, test cases are designed to cover all the functionality and performance, and to verify whether software functionality and performance and other characteristics are consistent with the user's requirements.

V-model is the most typical test model in order to test the embedded software to improve the performance and the effectiveness of the software.The V-model reflects relations between the testing process and design process.It shows the proper sequence of the testing process from left to right.

The testing of embedded software is different from the testing of general software.The embedded software testing includes the testing based on host environment and the testing based on target environment.

The host based testing involved the mbedded software unit testing and embedded software integration testing.In unit testing smallest design of software is test in order to provide the accuracy .In integration testing the all the modules having passedunit test will integrated through outline design specificationand detailed design specification.

The target based testing involved the embedded software validation testing and embedded software system testing.In embedded software validation testing test cases are designed to cover all the functionality and performance.During embedded software system testing the coverage isanalyzed to verify whether the coverage for function andperformance is complete ,at the same time the use of memory is dynamically monitored to prevent the memory leak.

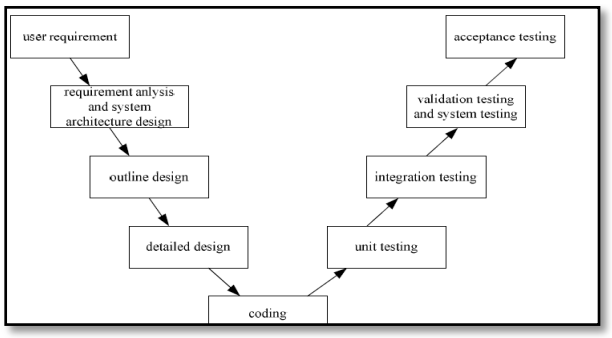

Figure 5.software testing V model

In 2009, Zhen LIand Bin LIU et al,[8] put forward an integrated framework to test embedded software with formal validation and automated test cases generation.The test case can be generated based on (Architecture Analysis and Design Language)AADL system component modes to support embedded software testing [9].The information of AADL system component modes and mode transitions can be captured to build a component test model.

In 2011, Johannes Kloos et al,[10] discussed the model-based testing of embedded system.The approach use the results of Fault Tree Analyses (FTA) during the construction of test models, such that test cases can be derived, selected and prioritized according to the severity of the identified risks and the number of basic events that cause it.

In 2010,Chorng-Shiuh Koong et al,[11] presents a supporting tool for testing embedded software. This tool can automatically generate test cases and test drivers, and supports unit test and coverage test which are based cross testing technology and multiple rounds mechanism.

3. OBJECTIVE

The ultimate goal of testing is to provide the organization with well informed advice on how to proceed – advice based on observed defects related to requirements of the system (either explicitly defined or implicitly assumed). The testing in itself does not directly improve the quality of the system. But it does indirectly, by providing a clear insight to the observed weaknesses of the system and the associated risks for the organization. This enables management to make better informed decisions on allocating resources for improving the quality of the system.

1) It is possible to test the various services provided by the particular embedded system.

2) Application programming interface will give the status of the services i.e. which services are working.

3) Proper interfacing of Application programming interface with the intended embedded system.

4) Application programming interface are designed to test the keyboard, printer ,Ethernet etc of the intended embedded system.

5) To perform the target based testing.

The reasons for testing are given below

1) If the defects are early detected,,it help to correct it easily.As the time passes the cost for fixing the defects goes on increasing.

2) High quality basic elements result into a high quality system. While the low quality basic elements,caused an unreliable system and which can't be solved practically by functional tests.

3) Defects detected during post development stages are difficult to trace back to source.

4) Defects detected during post development stages have to be corrected and this will lead to time consuming regression tests.

5) Good testing during the development stage has a positive influence on the total project time.

6) Straight testing of exception handling is only possible at unit level, at which the provision are made to individually handeled . Basically, quality cannot be added to the product at the end by Testing.

7) It should be built in right from the start.It is necessary to check the quality at every development stage by testing.

4. PROPOSED METHOD

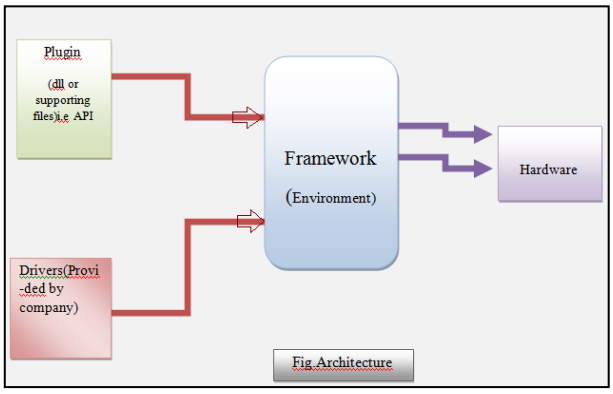

Figure 6.Diagram for Target based Testing

In target based testing the hardware device,framework and the connecting device are required to carry on the testing.The application programming interface(API) plays the role to caused the communication between the environment and hardware.Drivers are provided by the particular product manufacturer in order to have the compatabilitywith the environment.In this paper we are testing the USB device to have the required output with proper graphical user interface.The USB is used to have to test the sim card of any company.We are able to detect or test the sim card of mobile phone.It will help to display the information in a proper way than other cards.

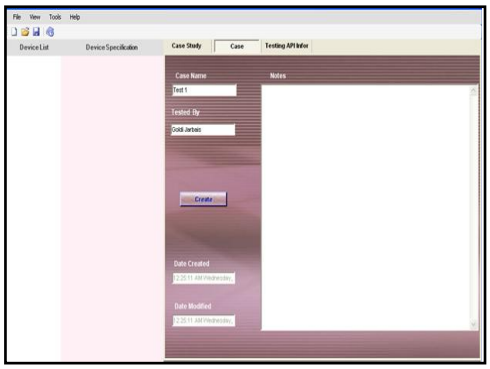

Figure 7.Graphical user interface for target based Embedded device Testing

The above figure shows the template for testing the sim/universal sim(SIMUSIM) for the mobile phone .With the help of API it is possible to testthe card of any phone.The case study is performed in order to have the proper sequence for testing.With the help of API,we are using the Case Name,TestedBy,Date Created and Date Modified field with in the template.

Case Name field is used in order to have a proper name to the test,the testing should be followed by the sequence to maintain the result.Tested By field is used to specify the name of the test performer who is going to perform the test on the simcard.The test will detect and test the card simultaneously i.e.it will check whether the card is within the USB device as the card get detected, the API will help to display the unique card number related to the sim.Date Created field is used to display the related date of the testing.We are able to get the present date and also the modified date with the help of Date Modified field.

The next part within the template is the Testing API Infor tab.Within testing API Infor,all the information related to the API are provided.It includes the information related to the communication plugins,protocol plugins,conversion plugins**.** Comunnication plugins are used to have the communication with the hardware and the environment in order to have connectivity.The protocol plugins are used to have the data extraction of the card through the way of communication plugins then to the protocol plugins and then last through the conversion plugins.It is also possible to displayed the result in desired form with the help of conversion plugins.

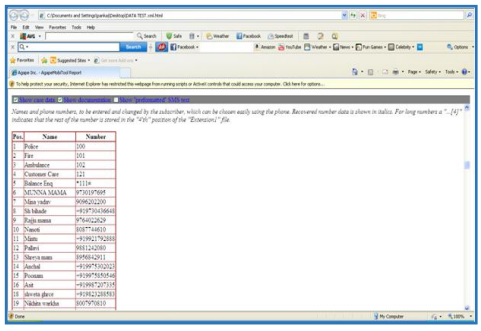

Figure 7.Result for Target based Embedded device Testing

The above figure shows the required output related to the card testing.It include displayed of the data in the form of xmlhtmlsheet.The name and the number of respective person get displayed in the form of sheet.The data can also be displayed in various form.

5.CONCLUSION

The middleware(generic) required device driver and operating system required target hardware for the execution.The device driver and target specific components related to any embedded system is executed if we provided with the target hardware.Other software components are depend on target hardware.So the target hardware is important.

Therefore the target hardware is used in order to test the application code of any software. Target-side is the actual target system to implement embedded system testing. When testing is in process, the module will collect related testing data and results and transfer to host-side for further analysis and presentation of test results as reference for next round of automatically generated testing data. During runtime, performance of how program is executed on target-side can be monitored.

4. REFERENCES

[1] Bart Broekman and Edwin Notenboo,"Testing Embedded Software",*TextBook,Pearson publication* ,2004

[2] AlperSen and EtemDeniz ,"Verification Test for Multicore Communication API.(MCAPI)",*2011 12th International Workshop on microprocessor Test and Verification*,2011.

[3] ZHU Zheng Wei And GUHao,"Design of the USB download interface based on embedded POS",*6thInternational conference on computer Science and education(ICCSE 2011)*,2011.

[4] Yang-Hsin Fan and Jan-Ou Wu," Middleware software for Embedded system",*2012 26th International conference on advance Information networking and application Workshop*,2012.

[5] Satoshi Hiratsukaand Akira Fusaoka,,"An On-Board Diagnosis Hardware for Embedded Systems", *1-4244-2368-2/08 2008 IEEE*,2008.

[6] Ahyoung Sung and ByoungjuChoi,"An Interaction Testing Technique between Hardware and Softwar in Embedded Systems",*Proceedings of the Ninth Asia-Pacific Software Engineering Conference (APSEC'02),* 2002.

[7] Hua-mingQian and Chun Zheng ,"A Embedded Software Testing Process Model", 978-1-4244-4507-3/09/2009 IEEE, 2009.

[8] Zhen LIand Bin LIU, Ning MA, YongfengYIN ,"Formal Testing Applied in Embedded Software", 2009.

[9] Xiao-li LU and Yun-wei DONG, Bo –SUN ,"Research of Embedded Software Testing Method Based on AADL Modes", *978-1-61284-486-2/111,2011 IEEE*,2011

[10] Johannes Kloos, TanvirHussain ,Robert Eschbach ,"Risk-based Testing of Safety-Critical Embedded Systems Driven by Fault Tree analysis" , *2011 Fourth International Conference on Software Testing, Verification and Validation Workshops*,2011

[11] Chorng-Shiu Koong1 Hung-Jui Lai1, Chih-Hung Chang, William C. Chu, Nien-Lin Hsueh, Pao- Ann Hsiung,ChihhsiongShih,Chao- Tun Yang" ,Supporting Tool for Embedded Software Testing, *2010 10th International Conference on Quality Software*,2010