www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 12 Dec. 2016, Page No. 19568-19573

Test Case Generation for Software Quality Assurance by Using PrT Nets

[1]M. Mahanandi Babu, [2]G. Ramesh, [3]G. Geetha
[1]M. Tech. Student, Dept. of CSE, JNTUACEA, Ananthapuramu
[2]Dept. of CSE, JNTUACEA, Ananthapuramu
[3]Dept. of CSE, JNTUACEA, Ananthapuramu
[1]nandibabu504@gmail.com
[2]ramesh680@gmail.com

Abstract: The quality of software is conditioned on checking and verifying of software. Programming testing performances fundamental early through of programming growth life cycle. It is the philosophy to finding absconds by executing our program without imperfections on a strategy of examinations and separating the faults outcomes and expected outcomes. For conveying the examinations, Information regarding the lead of system must be believed to in the first place, which can be taken from the models made in setup organize. Shortly a-days Model based testing for progressing structures is in impact. This paper acclimates an approach with convey test cases by utilizing PrT nets to adjust charge affirmation restrict with expansion time execution, other than the adaptability of the test code generator for nonstop frameworks. Petri Net (PrT) properties are assessed a brief time period later made trials made for check of Real time structure. It will in all likelihood redesign the capacity and meanwhile to decrease the high cost of the manual testing in setup organize itself.

Index terms- Functional testing, Model Based Testing, Petri nets, security testing, software assurance.

I. Introduction

Functional testing tells Quality Assurance (QA) process [1] and a category like black-box testing that built its trial with respect to the solution of the product segment under test. Functions to be tested by saving them input and looking the output, and internal program structure is structurally considered (dislike in white-box testing). [2] Functional testing usually orderly depicts what the system does. Functional testing varies from framework testing in that useful testing "confirms an edge work by checking it against.. design framework(s) on the other hand specification(s)", program testing "validate[s] a program by checking it against the disseminated customer or system prerequisites " (Kaner, discovered in 1999). Security testing is one procedure expected to uncover blemishes in the security instruments of a data framework that ensure information and keep up usefulness as proposed. Because of the consistent restrictions of security testing, passing security testing is not a sign that no imperfections exist and on that the framework enough fulfils the security prerequisites.

Software Assurance (SwA) is characterized as "the level of certainty that product is free from vulnerabilities, either deliberately outlined into the product or incidentally embedded at whatever time amid its lifecycle, and that the product capacities in the proposed manner."[1]

Functional testing does not suggest that you are attempting a methodology of your module or class. Functional testing tests the flaws on usefulness of whole framework. Intension of this paper is presently - a-days the use of Internet and portable figuring thusly raised our reliance on programming empowered frameworks. This dependence raises fundamental worries about programming immovable quality and security in light of the fact that an item frustration can incite to deplorable outcomes. An imperative means for ensuring the way of an item system is to test the structure with the desire to find potential disillusionments.

Petri net (generally known as a place/transaction net or PrT net) is a couple of numerical display accent for the identification of blame for dispersed frameworks. A Petri net is a made bipartite outline, in which the inside focuses are moves. The bi facilitated bends tells which spots are pre-and additional post conditions for which moves. Two or three sources [1] express that Petri nets were made in August 1939 through Carl Adam Petri - at the 13 years of age with a definitive target of depicting logical methods. Model Based Testing (MBT) is a promising way to deal with oversee motorize test conveyed by utilizing models of a System Under Test (SUT). The showing method of MBT clears up test essentials. Not the same as system exhibiting test showing focuses on itemizing what ought to be attempted, as opposed to getting all structure rehearses. MBT can be powerful in blame location in light of the mechanized era and execution of numerous tests.

II. Related work

Model Based Testing [3] is outstandingly sensible kind of testing which offers easy to making the trials from the business pre fundamentals that are formally noted by a model [4]. An electronic test period method, called Model based Integration and system Test Automation (MISTA) [5] [6]. It can diminish a significant measure of testing workload by supporting unmistakable testing works out.

MISTA can create security tests to practice potential unreliable practices. A Model Implementation Description (MID) [5], [6] detailed description will provide for the MISTA. For recognize both control and data related necessities for utilitarian testing, get the chance to control testing, with string models we are using a state Petri nets as a piece of MID unnoticeable segments.

The Formal meaning of PrT nets are characterized as takes after [1] [2]:

Petri net (generally calls as a place / transaction set or PrT net) is a few scientific demonstrating suggest for the clarifying of edge works. A Petri net is an organized bipartite graph, in which the hubs clarifies moves (i.e. events that may happen, suggested by bars) and places (i.e. conditions, implied by circles). The organized roundabout section which spot are pre and in addition post conditions for the moves (spoke to by bolts). A couple sources [7] express that Petri nets were produced in August 1939 by means of Carl Adam Petri at 13 years of age for the reason of Explaining substance systems.

Definition 1:
A Petri net N is a six-tuple (P, T, F, K, W, IN) where
$P = \{p_1, p_2, \ldots, p_n\}$ is a set of places
$T = \{t_1, t_2, \ldots, t_n\}$ is a set of transitions
F= is a set of normal arcs from P to T
K= capacity of each place. K(p) . k: p →N∪{w}
W=which attaches a weight to each arc.

Where IN: (P × T) → N is an input function that defines directed arcs from places to transitions and were OUT: (P × T)

→ N is a result framework that tells directed arcs from T to P.

If IN $(p_i, t_j) = k$, where $0 \leq k \leq 1$ is an positive integer, a represented arc from place p, to transition t_j is drawn with mark k.

If IN $(p_i, t_j) = 0$, no any arc is lined from p_i to t_j . in the same way, if OUT $(p_i, t_j) = k$, a direct arc is added from transition t_j to place p_i with k = 1. If k = 0, no any arc is added from t_j to p_i

III. Proposed work

The method of fault detection by using PrT nets, which is suitable for real time products and also large systems. This method is created for displaying the framework as Petri net, failures with moderate time constants is noticeable. The strategy is restricted to the recognizable of faults with PrT nets. Petri Nets are Test Models. PrT set in MISTA contains of conventional PrT nets [2] all weights in each formal total of tokens or curved names.

- **C = (P, T, I, O, F)**

  Places
  P = { p1,p2,…,pn}

  Transitions
  T = { t1,t2,…,tn}

  I/P
  I : T → P^r (r = total number of places)

  O/P
  O : T → P^q (q = number of places)
  F is a function

DOI: 10.18535/ijecs/v5i12.56

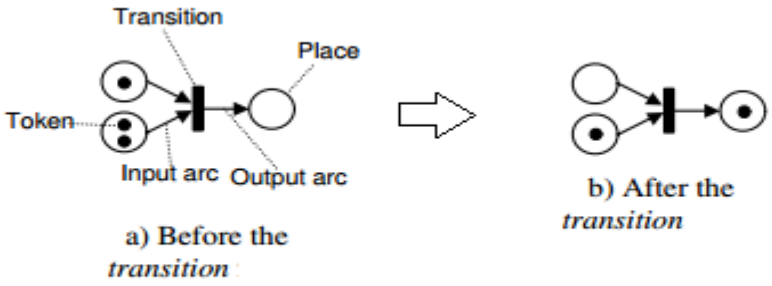

Fig1. Represent the working of PrT nets.

A few reviews present these pursuit strategies. The pursuit procedures go about as a middle part between the structures and the blame area.

1. Fault detection by using Prt nets:

The fault recognition by prt nets dynamic estimation signals which is reasonable for bigger frameworks is exhibited. By displaying the test case as a PrT net, disappointments with direct time constants will be noticed. The method is constrained to the distinguishing proof of sensor or process mistakes which are showed in signs identified with physical preservation amounts. After a fault is distinguished visualization without bounds framework's conduct can be given. The technique is connected to an atomic power plant optional cooling circle. The points of interest and downsides are talked about in detail.

1) MODEL : This contains description of variables
2) MIM : This contains logic and class name

Fault simulation is similar to testing process it evaluates fault detection capabilities of arrangement of test patterns. by coverage of each place detected faults are compare with model output response to known good output response For a given model of input test patterns. This MID contains of detailed information of model to be test with the conditions and logics. A MID detail involves a test show (PrT net),[2] and a MIM portrayal. The past does not use the use unobtrusive components of the SUT, however the keep going relies on upon the test appear and the SUT. In this section, we mainly present PrT nets and MIM, and after that delineate the running case to be used all through this paper.

In MIM specification for a test model mainly consists Test cases, local code, test suite and test drivers. From a testing model according to a chosen coverage criterion. The test sequence is developed as a transaction it represents detecting faults. From the initial place to all the places are checked by the transaction each transaction is Prt net is covered by at least one fault sequence in the model. Its main feature is that it is generating , which means that it provides a mechanism for defining own types of P rT nets called fault detection Petri nets .

DOI: 10.18535/ijecs/v5i12.56

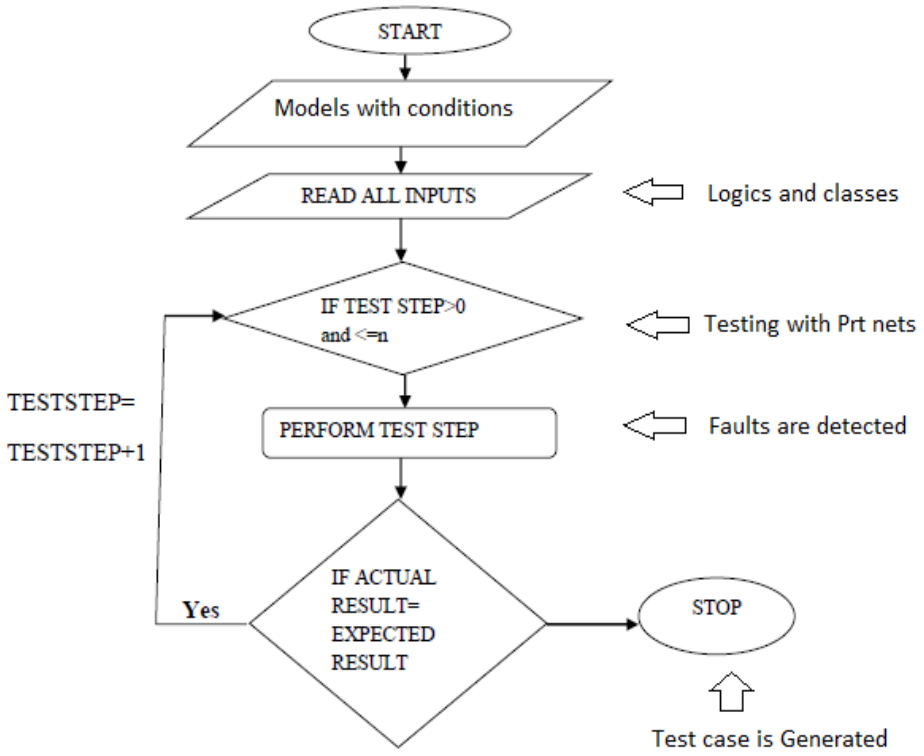

Fig2. Test case Generation process without faults by using prt nets.

In MIM assurance for a test exhibit generally contains Test cases , neighborhood code , test suite and aircraft testers. from a testing model according to a picked scope measure. The test gathering is made as a trade it addresses perceiving insufficiencies. From the hidden spot to each one of the spots are checked by the trade each trade is Prt net is secured by no short of what one accuse gathering in the model. Its rule highlight is that it is creating , which infers that it gives an instrument to portraying own sorts of PrT nets called fault area Petri nets .

The fig 2. Tells experiment Generation without blame by utilizing the Petri nets It is utilized to take care of the mind boggling issues like era of effective test suites for Generating experiment device. To recognise the faults at framework, pseudo-comprehensive testing is utilized; it has minimal test suite measure for output area when compared to with thorough testing of output . Test suites are advanced with seeded experiments, identified with limit value investigation of the input to contribute result. It decreases the accumulating time of source code, since it can run straightforwardly by MISTA translator. Test result demonstrates that, this strategy can create high branch scope result and diminished the pursuit space of the issue when comparing with client manual. As said in the above portrayal whole information is read the quantity of steps and the read the all input information values to contrast the fault location detection with the model. Until the without faults and then if actual result equal to the expected results. Every place is tested for the faults. Automation testing tools are used to reduce the manual power in unskilled and repetitive tasks.

To build a software system, making of appropriate tool is very important and a developer can do it. It covers the path entire conditions for functional testing diagram and utilizing white box testing. Thought about execution assessment of automation testing instruments. Here, they contrasted whole testing apparatus and selenium device. Other than the mix of various computerization mechanical gatherings, Selenium instrument is best known test suites which give analyzers with various structure to various investigations. It is an open source and fundamental testing gadget to test applications. It gives a sorted out made work selenium test suites. It supports test computerization on driver structure of utilizations transversely space. The experiment awards clients for empower acknowledgment, similitude and useful testing for more applications. Entire testing instrument mechanical assembly is an opened source automatic testing apparatus. It is one library for code dialect which drives the program. It requires programming limits. It is not a record and playback instrument like selenium. They have surveyed and considered the model tenets of this testing instrument. From their presented close results, it is obviously that selenium test is faultless to that of better testing instrument, where the selenium web driver is better choice in various conditions like usage of range particular tongue and framework.

DOI: 10.18535/ijecs/v5i12.56

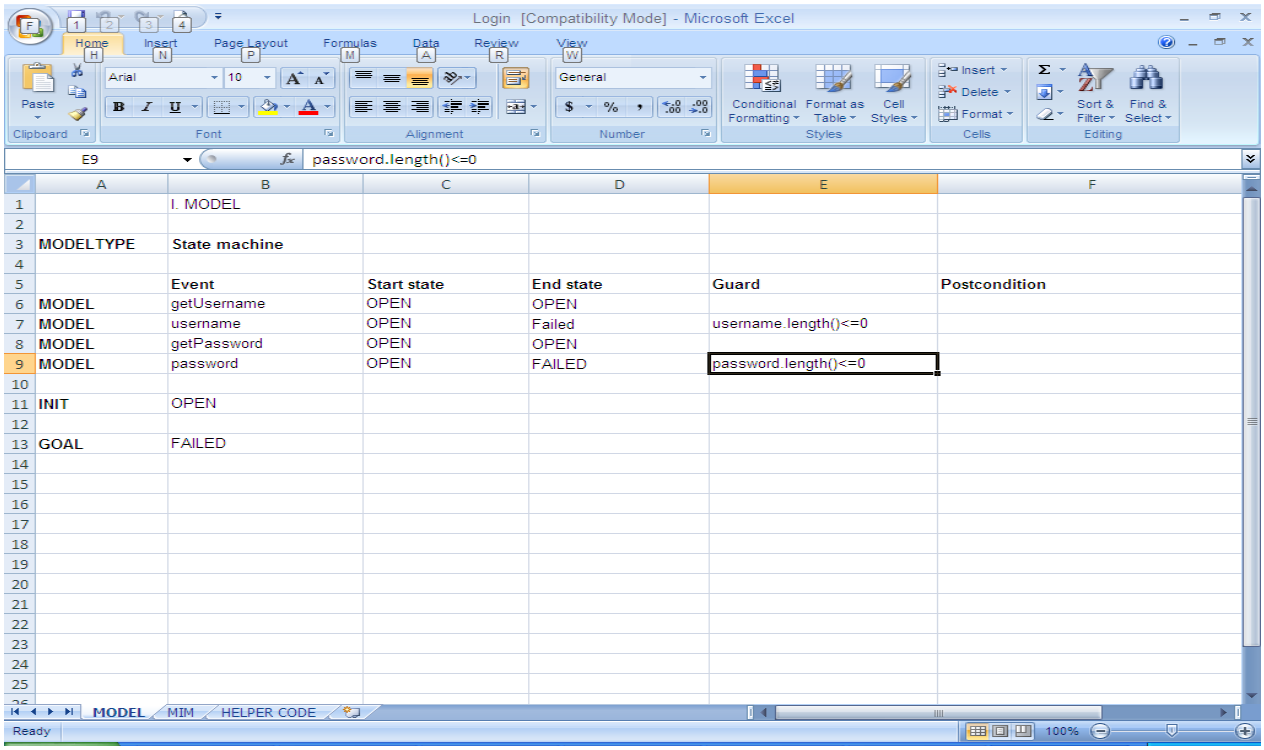

Figure 3: MIM

The above screen shot will describe model and its conditions and its descriptions.

IV. Result

As shown in the figure 3 experimental outcomes are gotten by the given data to draw the amount from ATM. Initially checks the account is opened or not and then checks for the availability of required amount in the account, if not available required amount it shows insufficient cash is your account else continues to draw the amount and close the transaction. Here every place is tested with the petri nets detects faults and then execute test case generation.

V. CONCLUSION AND FUTURE ENHANCEMENT

We have shown an accuse recognizable proof by PrT nets and after that create test cases thusly . Our approach has tended to the differing properties of PrT net and we can reason that recognizing the blemishes of an ATM structure is alright for use or not. Such system can contain imperfections and we can recognize them by demonstrating honest to goodness commitment to it. The consequences of frameworks, figuring, test criteria proposed for trial making is tremendous, in this paper the key considerations of computerization testing have been mulled over. These reviews address the current testing approaches and were used to further check the fundamental based testing. The change of the computations to add up to up obliged PrT nets can be push ahead.

References

[1]. J. Zender, I. Schiefewrdecker, and P. Mosterman, Eds., Model-Based Testing for Embedded Syst. Boca Raton, FL, USA: CRC Press, 2011.

[2]. M. Utting and B. Legeard, Practical Model-Based Testing: A Tools Approach. San Francisco, CA, USA: Morgan Kaufmann, 2006.

[3]. H. J. Genrich, "Predicate/transition nets," in Petri Nets: Central Models and Their Properties. New York, NY, USA: Springer, 1987, pp. 207–247.

[4]. D. Xu, "A tool for automated test code generation from high-level Petri nets," in Proc. 32nd Int. Conf. Applicat. and Theory of Petri Nets and Concurrency (Petri Nets 2011), LNCS 6709, Springer-Verlag, Berlin, Heidelberg, Germany, Newcastle, U.K., Jun. 2011, pp. 308–317.

[5]. D. Xu, L. Thomas, M. Kent, T. Mouelhi, and Y. Le Traon, "A model-based approach to automated testing of access control policies," in Proc. 17th ACM Symp. Access Control Models and Technologies (SACMAT'12), Newark, NJ, USA, Jun. 2012.

[6]. D. Xu, M. Tu, M. Sanford, L. Thomas, D. Woodraska, and W. Xu, "Automated security test generation with formal threat models," IEEE Trans. Depend. Secure Comput., vol. 9, no. 4, pp. 525–539, Jul./Aug. 2012.

[7]. D. Xu and K. E. Nygard, "Threat-driven modeling and verification of secure software using aspect-oriented Petri nets," IEEE Trans. Softw. Eng., vol. 32, no. 4, pp. 265–278, Apr. 2006.