International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
Volume 2, Issue 4, April 2013 **ISSN 2319 - 4847**

Implementation of Regression Testing Using Fuzzy Logic

Harsh Bhasin[1], Shailja Gupta[2] and Mamta Kathuria[3]

[2] M. Tech Scholars YMCAUST, IT Department
Sector – 6, Faridabad, India

[3]Associate Professor, YMCAUST, CE Department
Sector – 6, Faridabad, India

ABSTRACT

Regression testing calls for selection of appropriate test cases after modification have been made in the software. The task becomes difficult when the source code is not available. The work presents the use of fuzzy logic to develop an expert system, capable of selecting the test cases retaining the effectiveness and efficiency while at the same time reducing the number of test cases. The work is a continuation of an earlier work and the results obtained are encouraging.

Keywords: Expert System, Regression Testing, Software Agents, Test Case Prioritization.

1. INTRODUCTION

During its life span software undergoes many changes. These changes are done either due to new requirements or while correcting an existing module. When changes are made all the previous test cases must also be re-executed. This is neither physical nor possible. So, some of the test cases are selected in order to insert confidence in the software. This reduction in test case suite can be done by selection, minimization or prioritization. Selection selects the test cases relevant to changes. Prioritization prioritizes the entire test case suite so that most important test cases can be executed and minimization reduces the number of test cases in order to accomplish the task.

The process of regression testing can either be accomplished by conventional techniques or by machine learning techniques. Many works have been done as far as conventional techniques but a little has been done as per machine learning techniques are concerned. The work examines fuzzy logic as a premise of regression test prioritization technique. The work is based on one of our earlier works [1]. The paper presented henceforth presents verification of the technique presented by us and proves that fuzzy logic can be a base for expert system which is capable of test case suite prioritization. Fuzzy logic based systems have a human like capability to take decisions. The fuzzy logic based systems are more robust as compared to the standard if else constructs. The concept also incorporates a blend of nature to the system.

The rest of the paper has been organized as follows. The second session gives the literature review. The third section presents the background of the work. The fourth section examines the work and the last section concludes.

2. LITERATURE REVIEW

In order to understand the complexities of the system and explore the work done in the domain, an extensive literature review has been carried out. Three papers were selected which forms the background of the work. The review also considers an earlier work by us in the domain.

The earlier attempts of regression testing were studied [8-13]. The work by Zhiwei Xu[2] proposed a fuzzy expert system which makes use of classes, design specifications and rules and stores the rule firing history and infer the explanations. The inference engine in turn has an agenda and procedures to derive conclusions. The inference engine of the work used forward chaining and the active rules are stored in agenda.

The rules were executed until the agenda was empty. The work used schedule factor which explores the defect fix time and test case pattern. The defect impact factor cover test coverage factor, cover component availability and system setup constraints are used in order to generate the system test plan. Each test case is assigned a priority between 0 and 1 in order to generate a fuzzy set which is followed by the process of defuzzification.

The above work was tested on GSM system consisting of 29,748 test cases and 9805 executions. However, the escaped defect impact analysis is not considered in the work.

The work by Christopher Malz [3] proposed an adaptive test management system in which the software module agent determines the importance of the test case to that module. This task is accomplished by complexity of the software module, the number of faults, the number of changes, workload and the number of changes in other module. The rules are then generated which helps in assigning priority to the test cases. The local priority of each test case is determined which helps in determining the global priority. The work is verified on software for an automobile industry.

Another work by Christopher Malz [4] includes software module agents and test case agents where the software module agents are inquired about the perspective importance of test case by test case manager. The work is an extension of the work described earlier.

The above review helped us to find the gaps in the system and propose a theoretically sound technique. The above review along with our previous works [5-7] on regression testing helped to craft a technique which is presented in the following section.

3. BACKGROUND

The Architecture of fuzzy expert system has already been proposed [1]. Knowledge based fuzzy expert system consists of a test case evaluator, a local priority evaluator, a global priority evaluator, a software module importance estimator, a test case repository, an inference engine and a user interface. Local priority evaluator component calculates the local priority of the module by using the information obtained from Software Module agents and available databases. Software module importance estimator tells the importance of different software modules. Test case evaluator takes different test cases as input. It then determines test case importance of the test cases. These test cases are provided to local priority estimator by different test case evaluators. The concept of Coupling and cohesion is used by local priority estimator to determine local priority. Global priority evaluator determines the global priority. It is calculated as a weighted average value of local priorities.

Test case repository stores all the test cases that are generated. Inference engine generates the test importance by using global priority estimator, software module importance estimator and test case repository. In the work proposed earlier, a test case may cater to many modules. The local priority estimator evaluates the local priority of a test case. A set of local priorities help to estimate the global priority. The importance of module is also estimated. The software module important estimator and the global priority estimator helps the inference engine to generate the output. Figure 1[1] shows the architecture proposed in one of the earlier works.

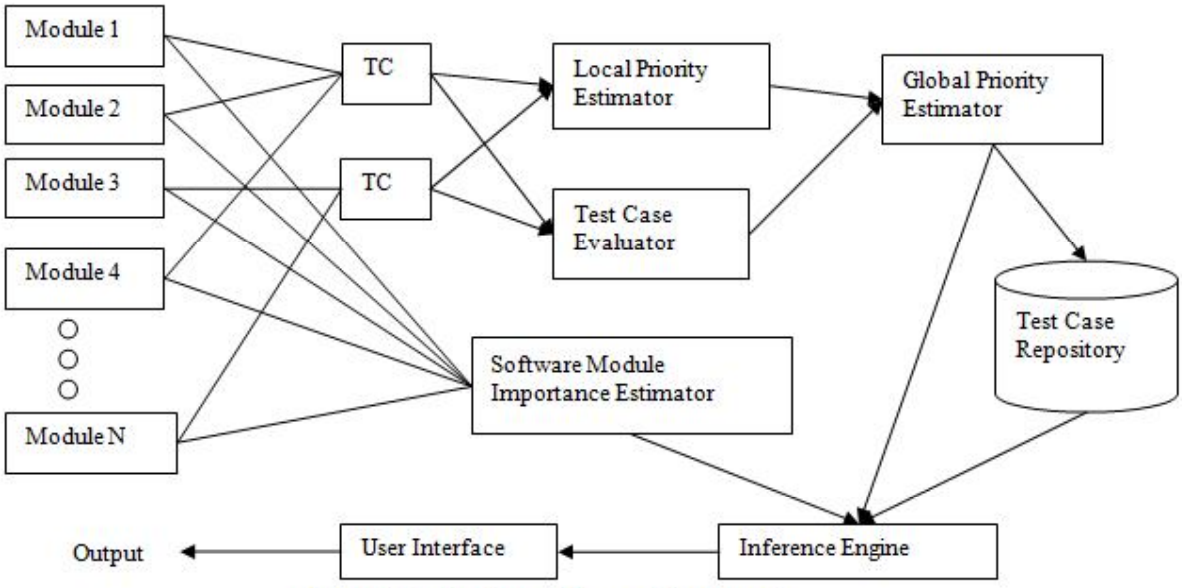

Figure 1 Architecture of Proposed Framework

4. IMPLEMENTATION

Data structure Used:

1. Comp_module[ ]: This data structure is used to store the complexity of each module.
2. Num_faults[ ]: This data structure takes into account the number of faults present in each module that are observed in previous test cycle.
3. Num_changes[ ]: it stores the number of changes in each module and is depicted by 1D array

4. Rel_imp_change[,]: It is a 2D array which contains the relative importance of each change on the scale of 1 to 10, and stored in the cell which is the intersection of module number and change number. 1 depicts the least importance of change and 10 depicts the highest importance.

The previous work considers workload as one of the important factors which has been deliberately removed, as it is an abstract quantity. Instead of this, the importance of changes has been added. It may be noted here that not every change is of equal importance. Therefore, the number of changed module a test case caters has also been taken into account. So, the local priority is now defined as

Local Priority = k1 * test importance + k2 * number of change module a test case caters

Where k1 and k2 are the constants that are stochastically defined. The formula for the global priority remains same as the previous work.

$$GP_x = \frac{(\sum_{i=1}^{n} LP_{xi} TI_i)}{\sum_{i=1}^{n} TI_i}$$

Where, Weighting factor $TI_i \geq 0$

n = number of software modules

5. CONCLUSION AND FUTURE SCOPE

The above work is implemented in C#. .NET 4.0 and is being tested against an Enterprise Resource Planning (ERP) system developed by Sahib Soft. The ERP system is a financial management system containing 53 modules. 3 cycles of test have been carried out. The results obtained so far are encouraging. The test cases have been manually generated. It is intended to generate the test cases by the Cellular Automata (CA) and Artificial life techniques proposed [6, 7] in the next paper. The proposed system is sure to be a milestone in the history of the discipline. The complete testing of the system is expected to replace the existing conventional Regression Test Selection (RTS) systems. In the next phase, 20 programs, selected by a group of lecturers and professors, will be tested.

References

[1] H. Bhasin, S. Gupta, M. Kathuria, “Regression testing using fuzzy logic”, International Journal of Computer Science and Information Technology (IJCSIT), 4(2), pp. 378-380, 2013.

[2] Z. Xu, K. Gao, T.M. Khoshgoftaar, "Application of fuzzy expert system in test case selection for system regression test", in Proc. IRI, pp. 120-125, 2005.

[3] C.Malz, N. Jazdi, P. Gohner, “Prioritization of Test Cases Using Software Agents and Fuzzy Logic", IEEE Fifth International Conference on Software Testing, Verification and Validation, pp. 483-486, April 2012.

[4] C. Malz, P. Göhner, "Agent-based test case prioritization”, IEEE Fourth International Conference on Software Testing, Verification and Validation Workshop, pp. 149-152, 2011.

[5] H. Bhasin, Manoj, “Regression Testing using coupling and genetic algorithm”, International Journal of Computer Science and Information Technologies (IJCSIT), 3(1), pp. 3255 – 3259, 2012.

[6] H. Bhasin, N. Singla, S. Sharma, “Cellular Automata based Test Data Generation”, Communicated.

[7] H. Bhasin, Shewani, D. Goyal., “Test Data Generation using Artificial Life”, International Journal of Computer Applications (IJCA), 67(12), pp. 34-39, April 2013. Published by Foundation of Computer Science, New York, USA.

[8] G. Rothermel, R. H. Untch, C.Chu, M. J. Harrold, "Prioritizing test cases for regression testing", IEEE Transactions on Software Engineering, 27, 2001.

[9] C. Malz, N. Jazdi, "Agent-based test management for software system test", IEEE International Conference on Automation Quality and Testing Robotics (AQTR), pp. 1-6, 2010.

[10] B. Broekman, E. Notenboom, "Testing embedded software", 2003.

[11] M. Wooldridge, N. R. Jennings, "Intelligent agents: Theory and practice", Knowledge Engineering Review, 10(2), pp. 115-152, 1995.

[12] H. Mubarak, "Developing flexible software using agent-oriented software engineering," IEEE Software, issue no. 5, 2008.

[13] F. Bellifemine, G. Caire, D. Greenwood, "Developing multi-agent-systems with JADE," John Wiley & Sons, Ltd, 2007.