ISSN:0975-9646

Regression Testing Using Fuzzy Logic

Harsh Bhasin[#], Shailja Gupta[*], Mamta Kathuria[*]

[#]*Department of Computer Science, Delhi Technological University Delhi, India*

[*] *YMCAUST, CE Department,Sector – 6, Faridabad, India*

***Abstract*— Regression testing is an immensely important process in the maintenance phase. The prioritization of test case becomes all the more important owing to the fact that it is not feasible to run all the test cases after each and every change. The proposed work dwells on the power of fuzzy expert system to make decisions which are better than the normal expert systems. The technique uses concepts like defect impact and coverage for test case selection. The work intends to propose a new framework for a fuzzy expert system based regression testing technique. The proposed architecture enhances the existing fuzzy logic based technique and removes faults therein. The proposed system architecture has been designed and initial results are encouraging. The system is poised to prove pivotal in the discipline.**

***Keywords*—Regression testing, prioritization, fuzzy expert system, coverage, defect impact.**

I. INTRODUCTION

Regression testing means re-execution of test cases after some modification have been made in the software, to ensure proper functioning of previous version's modules as well as the new ones. It gives assurance that newly added features do not cause any problem or side effects in the functioning of the system. This is generally performed in maintenance phase of software development cycle.

There are three types of regression testing techniques namely selection, prioritization and minimization. Selection deals with selecting the test cases that can be helpful in getting maximum number of errors if made to run, from all of the test cases. Its purpose is to identify the test cases that are relevant to some set of recent changes. Prioritization deals with prioritizing test cases to increase the rate of fault detection. Prioritization is used when the time for the testing is limited. The more important cases are tested so as to attain maximum coverage. Minimization deals with eliminating redundant test cases in order to reduce the number of test cases.

The main reason for performing regression testing is to ensure that changes in one part of the software do not affect other part.

Fuzzy Expert System consists of fuzzification unit that converts crisp values into fuzzified input. It consists of inference engine that contains if then else rules and a defuzzification unit to convert the result in a readable form.

Regression testing using fuzzification has been conceptualized by a few researchers as per the literature review. Fuzzy expert system is selected as the decisions made by it are better than the normal expert system [3]. The work improves the existing fuzzy logic based technique and removes the faults in it. The proposed system architecture has been designed and initial results are encouraging. Fuzzy expert system provides a better way of prioritizing the test cases.

The rest of the paper is organized as follows: Section 2 provides a brief overview of techniques proposed. Problem of the existing system is described in Section 3. Section 4 describes the architecture and Section 5 describes proposed work. Section 6 presents the results and conclusions.

II. LITERATURE REVIEW

An extensive literature review is carried out in order to understand the existing methodologies. It was found that, not much work has been done in the domain. However, some of the works that were studied have been summarized in this section.

Prioritization of test cases can be done by gathering test execution information [4]. Various techniques have been proposed to calculate cost and benefits of these techniques [5]. Theoretical and practical issues are discussed for the design and construction of intelligent agents [10]. Adaptive test management system has also been developed to increase efficiency [11]. KEE system has been extended to include context system and truth maintenance system [12]. A case study to test the accuracy of software quality module over much release has also been conducted [15]. A study on software quality estimation using SPRINT decision tree algorithm has been studied in order to accomplish the task of optimal decision making [14].

In the work carried out by Cristoph Malz a concept for an automated prioritization of test case using software agent and fuzzy logic has been presented [1]. The prioritization system determines the prioritization order which increases the test effectiveness and fault detection rate. For this, an agent based concept for prioritization has been used. The agent based prioritization system was realized using agent framework JADE. The work uses a connection over an interface agent to the fault management tool Bugzilla. The test management tool Testopia has been used in the work. Information about changes of the software module is obtained by an interface to the change management tool Subversion.

Christoph Malz and Peter Gohner have presented an Adaptive Test Management System based on software agent which prioritizes the test cases on the basis of available information from test team and development teams about software system and test cases [2]. The architectural model is provided in XML format. For evaluation, the prototype uses data of a company of automobile industry to compare result of test cases when executed with ATMS and without using ATMS. Result found was, that without using ATMS, test cases were repeated after software changes and no fault was found.

Fuzzy expert system based applications have been developed based on electronic commerce [7]. Various fuzzy

methodologies were considered to demonstrate the usefulness and to derive new ideas [8].

In the work carried out by Zhewei Xu, Kehan Gao and Taghi M Khoshgoftaar, a fuzzy expert system has been proposed in order to select the test cases when information of the source code is not available to testers [3]. A study has been done on the sample data from the GSM system test database. It has been demonstrated that the fuzzy expert system is suitable for solving inaccurate and subjective problem as encountered by system test case selection. The system has also been used to manage the complexity of mobile payment service [6]. A structural and controlled study to test embedded software has also been studied in order to understand the fuzzy expert system [9].

III. Problems In Existing System

The detailed study of the existing methodologies helped in figuring out various problems thus facilitating the premises for proposing a new technique for the problem. The proposed technique aims at removing the limitations observed and provide the better result.

The previous work studied states that the high value of local priority indicates probability of finding faults but no formula has been stated to calculate the local priority. The SM agent gives the information about the change. TC agent answers the question how probable is the probability of finding faults with the test case and calculate its global priority that depend on local priority but both the techniques have not been theoretically justified. In the above work, test importance is an indication of need of high test intensity for software module and value of test importance lies between 1to 10.

IV. Proposed Work

The proposed architecture of Fuzzy Regression Expert System (FRES) consists of three components knowledge base, inference engine and user interface. Knowledge base contains all the rules. Inference engine takes the decision by checking which rules are satisfied by facts, prioritize the rules that are satisfied and execute the highest priority rules. The rules are to be prioritized based on premise discussed in the section. Inference engine processes the rules that are extracted and whose patterns are satisfied by facts in contention. The user interface presents the user available facts and other information as input.

A. Knowledge Base

Knowledge base consists of objects that can be rules, classes and instances or goals and tasks. It consists of domain knowledge that is useful for the inference engine. It consist natural language rules in form of if then else. Rules depict the knowledge to be used by the expert system. For example, the rule test case suite which has higher priority based on calculating the local priority and hence it would have higher priority

B. Inference Engine

Expert system searches for rules in knowledge base at first. Inference engine then relate different pieces of knowledge against relevant data. Example, as per the previous work, if coupling becomes high then local priority in software module becomes more. More is a fuzzy adjective and hence the value is calculated using fuzzy system.

C. User Interface

The user interface allows user to enter rules. It also presents explanation to conclusions and the data acquisition facility.

D. Software Module

As per the previous work, a software module is a basic unit of a program which accomplishes a particular task. A program is the combination of different software modules.

E. Software Module Importance Evaluator

Software Module Importance Evaluator evaluates the probability of finding the faults in a module by the requisite rules.

F. Test Case Evaluator

A test case consists of inputs and outputs. Set of test cases is called as test case suite. A module can have many test cases. The Test Case Evaluator component evaluates different test cases and indicates the importance of the test cases.

G. Local Priority Evaluator

Local priority is the probability of a test case to find faults in a software module that is related to the test case. To determine the local priority, the software agent (TC-agent) uses information from SM-agents and available databases. The local priority component calculates the local priority of module.

H. Global Priority Evaluator

Global priority is the priority of a test case with respect to all other test cases. The software agents (SM agents) uses test importance of a software module and local priority of test cases to determine global priority.

I. Inference Generator

Inference Generator is a reasoning generator that is based on some information that derives to a conclusion or new information i.e., from the given rules and facts, it derives new rules.

Diagrammatic representation of the proposed architecture is shown in Fig. 1.

A software program consist of various software modules here represented by module1, module2… module n. A test case evaluator (TC evaluator) considers various software modules and their related test cases. It takes different test cases as inputs and determines test importance of the test cases. These test cases are provided to local priority estimator by different test case evaluators. Local priority estimator takes the results of test case evaluator and obtains local priority by using coupling and cohesion among different software modules. Local priority is being obtained by calculating the weighted sum of the entire incoming and outgoing link from the module. Global priority is then calculated by TC-agent as the weighted average value of the local priorities.

All the test cases generated are stored in the repository called test case repository. Software module importance estimator interacts with different software modules to tell the importance of different software modules. Inference engine then using the global priority estimator, test case repository and software module importance estimator generates test importance. It is conveyed to user via user interface.

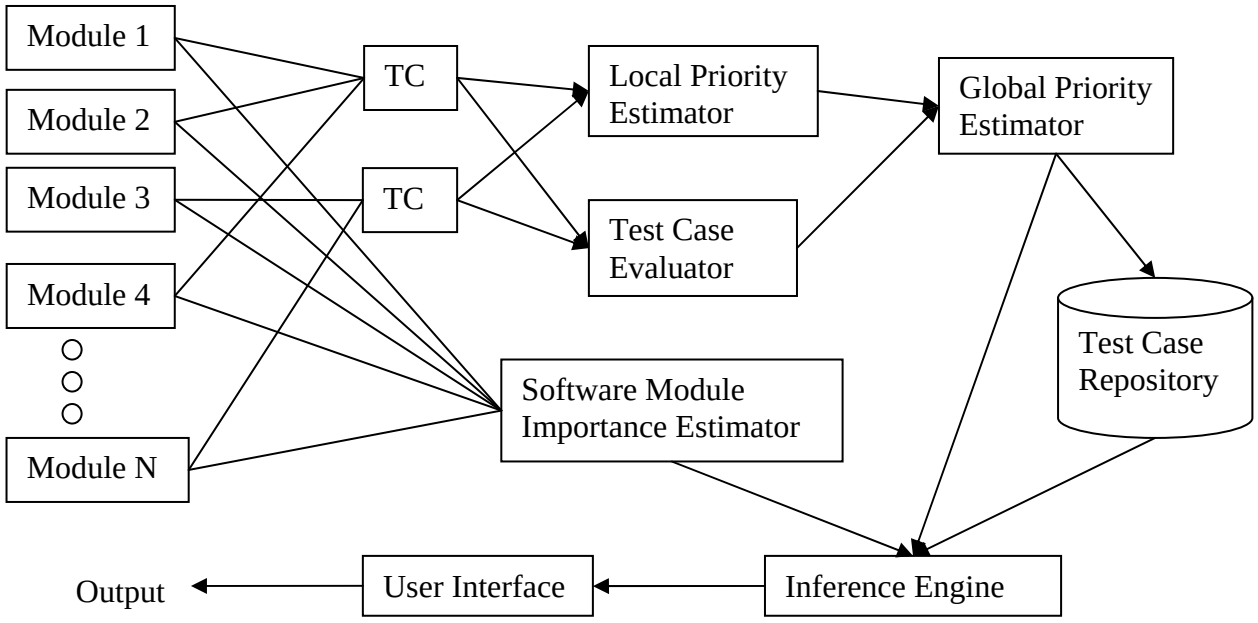

Fig. 1 Architecture of Proposed Framework

V. CONCLUSIONS

Regression testing becomes an important task in retesting of the whole system. Once a change is made it is not possible to retest all the test cases of the test suite as it will consume lot of time and is not a feasible solution. Therefore, prioritization of the test cases present in test suite becomes important. The regression testing using fuzzy system used here provides a new approach to prioritization of test cases. It has been found that using fuzzy expert system, provides better results than the other decision making systems. The work proposed is being implemented and analyzed. The results obtained so far are encouraging. The proposed work is to be tested using commercial software of 8000 lines of code. The software consists of around 200 modules and 500 test cases.

REFERENCES

[1] Malz, C., Jazdi, N., Gohner, P., "Prioritization of Test Cases Using Software Agents and Fuzzy Logic", Software Testing, Verification and Validation (ICST), 2012 IEEE Fifth International Conference on, On page(s): 483 – 486.

[2] Christoph Malz, Peter Göhner, "Agent-Based Test Case Prioritization," icstw, pp.149-152, 2011 IEEE Fourth International Conference on Software Testing, Verification and Validation Workshops, 2011.

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

[4] G. Rothermel, R. H. Untch and M. J. Harrold, "Prioritizing Test Cases For Regression Testing", IEEE Transactions on Software Engineering, vol. 27, no. 10, October 2001.

[5] C. Malz and N. Jazdi, "Agent –based Test Management for Software System Test", IEEE International Conference on Automation, Quality and Testing, Robotics, pp. 1 -6, May 2010.

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

[7] M.Schneidr, A.Kandel, G. Langholz, and G. Chew. Fuzzy Expert System Tools. John Wiley & Sons, New York, 3rd edition, 1996.

[8] Z. Xu. Fuzzy logic techniques for software reliability engineering. Ph. D. dissertation, Florida Atlantic University, Boca Raton, Florida, USA, May 2001.

[9] B .Broekman, E. Notenboom, "Testing embedded software," Addison- Wesley, 2003.

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

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

[12] R. E. Filman. Reasoning with worlds and truth maintance in a knowledge based programming environment. Communications of the ACM, 31(4):382-401, 1988.

[13] J. Giarratano and G. Riley. Expert Systems: Principles and Programming. PWS Publishing Company, Boston, 3rd edition, 1988.

[14] W. D. Jones, J. P. Hudepohl, T. M. Khoshgoftaar, an E. B. Allen. Application of a usage profile in software quality models. In Proceedings of the Third European Conference on Software Maintainance and Reengineering, pages 148-157, Amsterdam, Netherlands, Mar. 1999. IEEE Computer Society.

[15] T. M. Khoshgoftaar, E. B. Allen, W. D. Jones, and J. P. Hudepohl. Accuracy of software quality models over multiple releases. Annals of Software Engineering, 6, 2000. In press.