iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 3 **Issue:** X **Month of publication:** October 2015

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Automated Incoup Tool for Coupling Based Integration Testing

Ms. S.Shanmugadevi,M.Sc (IT), (M.Phil)
Assistant professor, Department of CS & IT,
Nadar Saraswathy College Of Arts and Science, Theni-625531

Abstract-Software testing is the major part in the development of software. Testing acts as the main work to identify the faults in the software product and provide errors to rectify. To ensure the software quality, errors are to be rectified with various types of testing such as unit testing, specification, validation, integration and other types of testing. Here the major research concentrated on integration testing part where it tests the interactions of different components resides in the coding part and functionality of the project. One of the major part in the integration testing is the coupling based integration testing that is depend on coupling relationships that exist among different variables across different call sites in functions. The existing research for test data generation deal only unit level testing and there is no process for test data generation for coupling based integration testing. In this paper, we have proposed a novel approach for systematic test generation for coupling based integration testing of programs designed with object oriented concepts using stream reader functions. Our method, allows the user to load the two divisions of class files, split the class files and then compare it for class availability. We have implemented a prototype tool INCOUP IN Dot Net and successfully performed different class codes experiments for the generation of test data. In experiments with this tool, our proposed method has given much better results as compared to random testing and E-Coup testing.

Keywords- Coupling way, Ensuing technique. Antecedent system, Coupling variable, coupling sort

I. INTRODUCTION

The process of executing the program with the intent of finding an error is called Testing. Software testing defines to detect as many errors as possible with minimum cost. Testing is not restrained only to the detection of error it also assists with the cost of the functional properties of the software [10]. Software testing is to ensure that the software meets all the requirements of the customer to check whether the product meets functional and performance objectives and to ensure safety and regulatory compliance for the production standards are met. It achieves zero-defect quality software but it is not possible in reality. During the software development it consumes the half amount of total cost involved. Integration testing is the type of software testing in which the each individual software modules are combined and tested to integrate. The integration is done after unit testing and before validation testing.

Software Testing is an examination directed to give acceptable stakeholders for majority of the data over the product or service under test. Software testing assumes a key part in improvement of software under software engineering. Test data generation is the most substantial and vital phases under software testing. Software testing is not conceivable deprived of suitable test data. Software testing perceives errors in software and confirms quality. Various types of testing such as Unit, Integration or system level can be performed by software testing. In those testing process, integration testing is used to tests the interactions of different modules, when they are incorporated together in explicit application, for the smooth functionality of software system. An integration testing approach is implemented as coupling based testing that is based upon coupling interactions that occur among different variables over diverse call sites in functions. Diverse sorts of coupling exist between variables across different call sites. There may be no worth for test data generation for coupling based integration testing.

In this paper, we have suggested a novel approach for automated test data generation based on object oriented programs using genetic algorithm for coupling based integration testing. In this approach, the coupling path acts as input which holds multiple sub paths and generates the test data using genetic algorithm. We have implemented a prototype tool InCoup in DotNet and it results in

©IJRASET 2015: All Rights are Reserved

effective manner by performing different experiments for the generation of test data. The proposed approach has much better results as compared to existing approaches.

II. BACKGROUND

A. *Testing In Object Oriented Testing*

Fault detection capabilities stated that object-oriented programs mainly focus on software units to the way software classes and components are connected. It found a less detection on unit testing and more on integration testing. The inheritance and aggregation relationships are combined with polymorphism, it introduces a new kind of integration faults, and by using the testing criteria it makes the effectiveness of the polymorphism and inheritance. The relative effectiveness of several coupling based object oriented testing criteria is explained through the set of experiments. It concludes that OO criteria are all not effective at detecting faults due to the use of inheritance and polymorphism than branch coverage [3]. Testing challenges explains the testing is one of the critical processes during software development life cycle. Improving the software quality plays an important role in the success of software product. Web-based applications are rapidly increasing the complexity by emerging and evolving the applications. Heterogeneous and the distributed components and applications along with their multi-platform support and cooperativeness make these applications more complex and increasing in the size. Quality assurance is becoming more crucial and important; testing is one of the key processes to achieve and ensure the quality of these software or Web-based products. Testing challenges on web based applications is more beneficial where as testing team performs the both integration and interoperability testing [4]. The integration testing is an important part of the testing process, but few integration testing techniques have been systematically studied or defined. In Coupling based testing criteria [21], the integration testing is to develop practical, effective, formalizable, automatable techniques for testing the components which are integrated. It provides an integration testing technique which is based on couplings between software components. It supports integration testing of software components, and satisfies the USA's Federal Aviation Authority's requirements for structural coverage analysis of software. Through this technique the test data are not automated. In this research the automation test data generation is used for coupling based testing [1].

Defects detection in object oriented programs leads to logical error are a burden for the user or programmer. The complier is not well equipped to track such defects. A piece of code can be tested to increase confidence by exposing potential flaws or derivations from user's equipments. The algorithm detects the defects automatically. The algorithm checks the data type of the actual parameters and formal parameters for an exact match. If a match doesn't occurs the tool report the situation. Further it is extended to detect the defects in c# programs caused by typographical mistakes and omission of characters which results in execution error. By this approach, the defect rate for the users of the class is reduced [9]. An imperative language such as c++ is a familiar object oriented programming that is widely used for reusability and increase ability to enlighten with other languages. Testing is not confined only to the detection of bugs; it also assists with the evaluation of the functional properties of the software. It includes the defects occur due to unintended characters, wrong usage of data member and formal parameter and a missing argument indicator in console applications. By this approach unit testing improves the quality of the code in terms of reducing programmer's burden, time and effort [8]. Automation on unit testing for java programs explains that program testing is consuming higher costs on the development process. Because of t the expensive in costs it is not frequently done well and results are not always satisfactory. Testing is the primary method to ensure that programs compile with requirements. It investigates the use of an evolutionary approach, called genetic algorithms. The genetic algorithm used for the test data generation and the use of program specifications and JML is used for the test result determination. A proof of-concept tool has been implemented and shows that a complete automation is feasible for unit testing in Java programs. Automated testing techniques such as manual testing by testing significant portion of object-oriented programs, as methods in object-oriented programs tend to be small manual testing can focus more interesting problems, e.g., inter-class testing [2].

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue X, October 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

III. PROPOSED APPROACH FOR INCOUP TOOL

In this approach mainly designed for four stages as shown below:

Load Programs

Split Coding

Split Classes and

Coupled Integration Check

Load Programs

In the Load programs the user is allowed to load the two java coding different programs with class file to check for coupled integration testing.

Split Coding

In the split coding process, each and every word of the code is tokenized and listed out in the list view control box. This process is used for both source code 1 as well as source code 2.

Split Classes

In the split classes process, available classes in each program is listed out in a separate register. Once the classes filtered out, it will allow for coupled integration check.

Coupled Integration Check

In the coupled integration check, the available classes in each list are compared together and check for class availability. The matched classes related to coupled integration testing whereas the classes which are unique are not coupled together.

A. Activities Of Incoup Tool

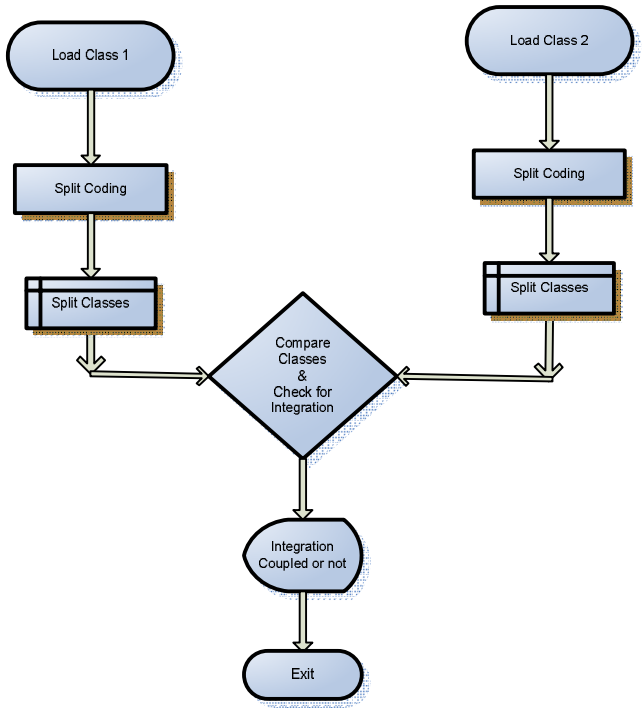


Fig 3.1 Activities of InCoup tool

The Fig 3.1 explains that it split the coding for the loaded java programs. The splitted coding is used for splitting the classes from

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue X, October 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

the loaded java programs. Then both the dividend classes are allowed for coupled integration checking by the tool. The classes which are called by one another that they are integrated properly and the unmatched classes are not properly integrated together. For coupled integration testing, the interface is viewed for the claiming variables preceding calls and returns initially utilization inside units and after calls. The INCOUP tool indicates the test information era stream to the proposed approach about test information era for coupling based integration testing.

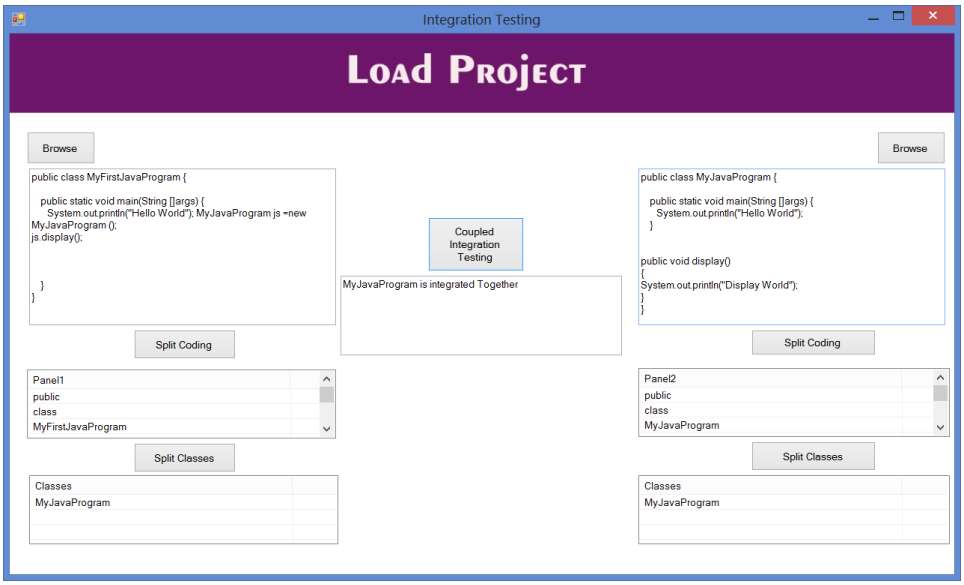


Figure 3.2 : InCoup Tool - Integration Testing Form

Consider the Figure 3.2, in the left hand side Class1 ‘C1’ is loaded and then the internal coding of C1 is split in the list viewcontrol which is denoted as List(C1) and then the classes from List(C1) is split individually which is denoted as Split(list(C1)). After C1 process completes, then the next class ‘C2’ is loaded and then the internal coding of C2 is split in the listview control which is denoted as List(C2) and then the classes from List(C2) is split individually like C1 which is denoted as Split(List(C2)) . Now the user allows to enter the compare the Split(List(C1)) with Split(List(C2)), the class names listed in the Split(List(C1)) match with the Split(List(C2)) are integrated together whereas the unmatched class names are not properly integrated together.

IV. EXPERIMENTAL RESULTS

Using INCOUP tool, a batch set of programs are allowed for coupled integration testing, they are as follows

No. of Lines of Coding		No. of Classes Identified		No. of Classes Integrated	No. of Classes not Integrated properly	Process Taken (in Secs)
Source1	Source 2	Source1	Source2			
1875	1456	189	241	51	124	95
2463	2896	279	319	16	219	113
15200	18630	1263	1165	521	326	361
25630	26893	2130	2529	682	335	608
51623	52698	4291	4380	1521	1450	1252
85963	96321	5852	5494	2747	1685	2086
126452	128632	11464	10719	5647	2781	3190

Fig 4.1 Table shows the batch set of programs allowed and

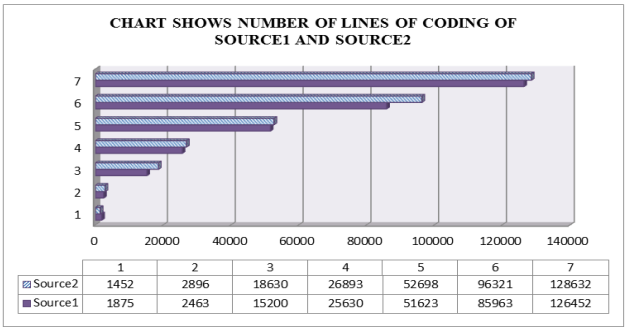


Fig 4.2 Chart shows Number of lines of coding of source1 the results performed for the batch and source 2

©IJRASET 2015: All Rights are Reserved

Volume 3 Issue X, October 2015
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

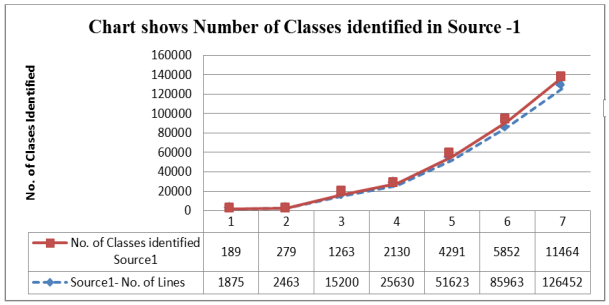


Fig 4.3 Chart shows Number of Classes identified in Source-1

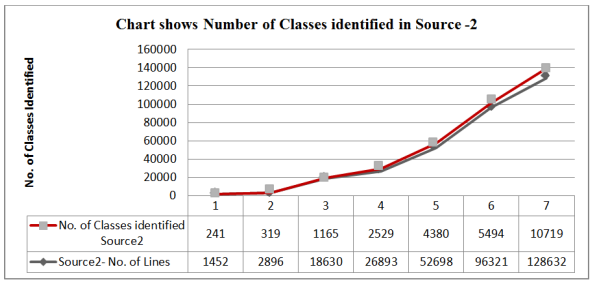


Fig 4.4 Chart shows Number of classes identified in Source-2

Total lines Evaluated	No. of Classes Integrated	No. of Classes Not Integrated	Process Taken (in Secs)
3327	51	124	95
5359	16	219	113
33830	521	326	361
52523	682	335	608
104321	1521	1450	1252
182284	2747	1685	2086
255084	5647	2781	3190

Fig 4.5 Combined Total lines evaluated and result performed

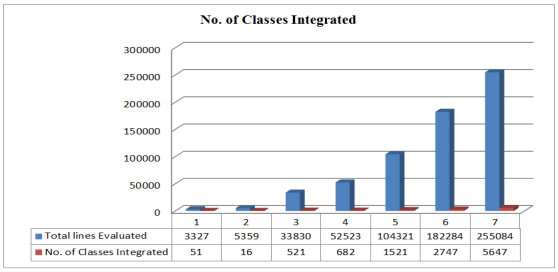


Fig 4.6 Chart shows the number of classes integrated

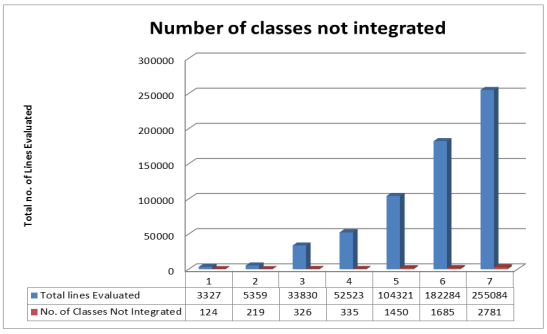


Fig 4.7 Chart shows the number of classes not integrated

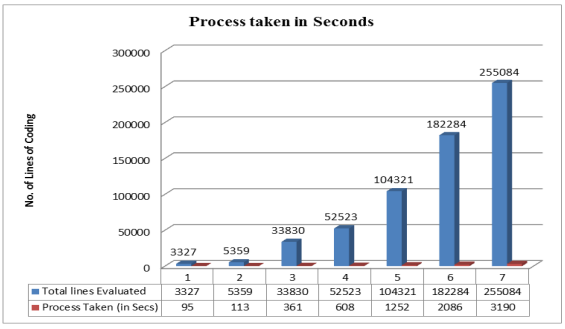


Fig 4.8 Time taken for a process in Seconds

V. CONCLUSION

A novel approach is recommended for the programmed test information era for coupling based mix testing. This recommended approach needs two phases. In the first phase, the coupling ways would identify the utilization and static examination of the project. The second phase tests the information era for coupling paths, as an identifier in principal phase and utilizing hereditary calculation. Hence a model is known as InCoup tool which used as evidence for the particular integration of the classes. InCoup is a superior tool to avoid the irregular test information. The results are analyzed and experimented using this InCoup tool which gives more effective results than the existing irregular testing. The tool InCoup is utilized to test information era same time it need a ways to aid gave manually. In future, we will improve InCoup for programmed test way era utilizing coupling data clinched along with the projects.

REFERENCES

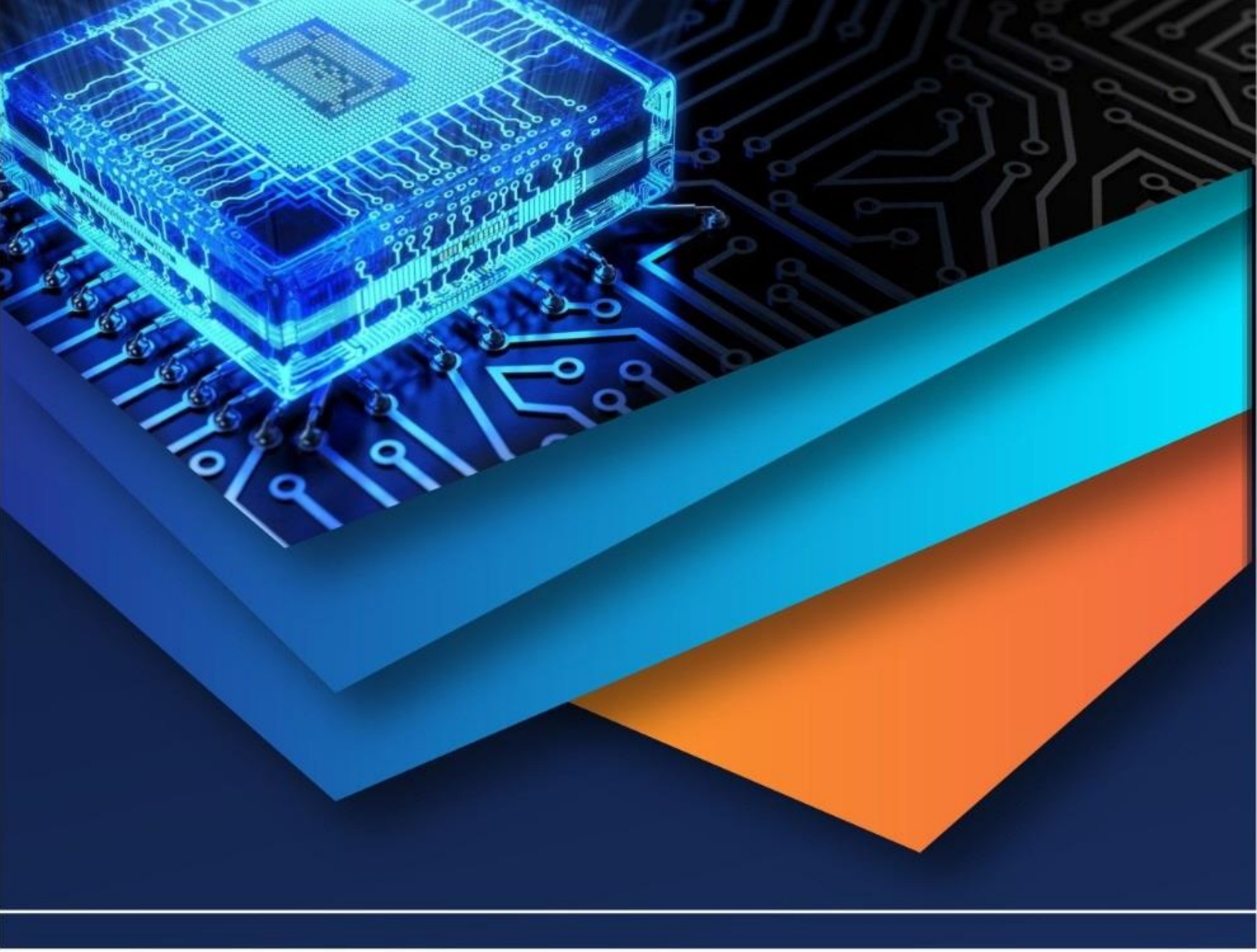
[1] Alberto Gonzalez-Sanchez. (2014). Testing Software and Systems. Built-in Data-flow Integration Testing in Large-Scale Component-Based Systems ?. 79-94 (5), 1-12.

[2] Zhenyi Jin and A. Jefferson Offutt, Coupling-based Crite-ria for Integration Testing. The Journal of Software Test-ing, Verification, an d Reliability, 1998.8(3): p. 133-154.

©IJRASET 2015: All Rights are Reserved

[3] W. T. Tsai, Y. Chen, R. Paul,N. Liao and H. Huang. (2004). Cooperative and Group Testing in Verification of Dynamic Composite Web Services.Proceedings of the 28th Annual International Computer Software and Applications Conference. 1 (04), 0730-3157.
[4] Wesley Klewerton Guez,Thelma Elita,Silvia Regina. (2014). Evaluating different strategies for integration testing of aspect-oriented programs.Journal of the Brazilian Computer Society 2014. 20 (9), 2-17.
[5] Sthamer, H., (1996) "The automatic generation of software test data using genetic algorithms", PhD Thesis, University of Ghamorgan, Pontyprid, Wales, Great Britain.
[6] Lionel C. Briand, Jie Feng,Yvan Labiche. (October 2002). Experimenting with Genetic Algorithms and Coupling Measures to Devise Optimal Integration Test Orders. Carleton University, Technical Report SCE-02-03 . 03 (1), 02-03.
[7] Zhenyi Jin, A. Jefferson Offutt. (1998). Coupling-based criteria for integration testing. 5. Software Testing, Verification and Reliability. 8 (1), 133-154.
[8] Michaela Greiler ,Gerhard Gross. (2009). Runtime Integration and Testing for Highly Dynamic Service Oriented ICT Solutions – An Industry Challenges Report. Software Engineering Research Group. 4 (4-6), 51-55.
[9] Lee Copeland, "A Practitioner's Guide to Software Test Design", STQE Publishing, 2004.
[10] Roger T. Alexander,Jeff Offutt,James M. Bieman. (2002). Fault Detection Capabilities of Coupling-based OO Testing. ISSRE. 2 (1), 12-15.
[11] Hong Zhu,Xudong He . (2005). A Methodology of Component Integration Testing. Springer Berlin Heidelberg. 1 (2), 239-269.
[12] Boris Beizer, "Software Testing Techniques", International Thomson Computer Press, 1990.
[13] Umar Farooq,Usman Azmat. (Jan 2009). Testing Challenges in Web-based Applications with respect to Interoperability and Integration.Department of Interaction and System Design. 1 (2), 25.
[14] Leonard Gallagher,Jeff Offutt. (2006). Test Sequence Generation for Integration Testing of Component Software. Oxford University Press on behalf of The British Computer Society. 1 (1), 208-299.

©IJRASET 2015: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)