Proc. of The Second Intl. Conf. On Advances In Computing, Control And Networking - ACCN 2015

ISBN: 978-1-63248-073-6 doi: 10.15224/ 978-1-63248-073-6-25

To measure the impacts on fault injection model in Component Based Software Development

Ms. Jyoti sharma Dr. Arvind Kumar

***Abstract*—Component based software engineering (CBSE) is gaining substantial interest in the software engineering community. A lot of research efforts have been devoted to the reliability, fault tolerance and complexity of components in CBSE. Testing the software component is an important approach which guarantees and enhances the reliability of components and as the reliability factor is directly related to the complexity of the components in CBSE. So automatically the complexity of the software will get improved. Merely selection of less complex and reliable components is not sufficient but we have to select the optimal components for the software system including the factor of fault tolerance. This paper proposes a testing strategy for finding the complexity of components and also a modified approach of fault injection model is proposed.**

***Keywords*—component testing;fault injection model;testing strategy**

I. Introduction

CBSE is an approach which is used to enhance the reusability because reusability is a way to improve efficiency and productivity of software systems. Component based software systems are mainly constructed from the reusable components such as third party components and the commercial off the shelf components (COTS). As the rapid increase of software system size and complexity, it is very important to reduce the high software cost, time and complexity while increasing reliability, Performance and quality so on.

Although there are many published articles addressing the software complexity and reliability in component based programs, very few papers address the problems and challenges of fault injection model or fault injection technique. To the best of our knowledge component testing in CBSE for the purpose of enhancement of reliability of system is rarely researched as a special subject. Presently there have been few testing approaches for components testing, which are mainly derived from traditional software testing approaches.

In related research on testing of components in CBSE, J.Chen [1] et al. presented a testing approach of component security based on fault injection. They discussed the testing approach with its requirement specification described by XML according to certain schema. They discussed the case study and its result by injection of various faults. C.Jin-fu[2] et. al. presented a fault injection model oriented testing strategy for component security. They have addressed the fault injection model in a different way with some new factors with a framework of environment fault injection. They have

presented a testing strategy of component security named fault injection model-oriented component vulnerability detecting (FCVD) with the requirement specification of component security with a detecting algorithm of component vulnerability. Han et al.[3] proposed the specifications of component security requirement which could enhance the testability of component security to some extent. In the paper we propose a testing approach of components in CBSE environment based on a new fault injection model with some factors added. The motivation of this paper is to introduce a simple and quick technique for determining the complexity of the components with a modified fault injection model.

II Testing Approach of Components Based on Modified Fault Injection Model

Fault injection is important to evaluating the dependability of computer systems and it is usually used in the software robustness testing and software error locating. Researchers and engineers have created many novel methods to inject faults, which can be implemented in both hardware and software.

A. *Testing Approach*

New Model of testing approach (MFCVD) includes seven elements TC, SR, DM, TD, NFIM, DA, and EE. (1) TC expresses tested component. (2) SR expresses the effective security requirement specification on tested component. (3) DM expresses the environment of dynamic monitoring for C. (4) TD expresses the testing driver for DM and TC. (5) NFIM expresses the new fault injection model. (6) DA expresses the set of detecting algorithms which includes Di, and Di is a detecting algorithm on security. (7) EE expresses the effective evaluation criteria for the tested component.

B. *Modified Fault Injection Model Oriented Component Vulnerability Detection(MFCVD)*

Fault injection has been widely used to evaluate the dependability of systems and to validate error handling mechanisms. This technique consists of introducing faults during an execution of system under test and then observing

Ms. Jyoti sharma
Ph.D. [CSE] Research Scholar
SRM University, Sonepat, Haryana, India

Dr. Arvind Kumar
Computer Science and Engineering Department
SRM University, Sonepat, Haryana, India

its behavior. By doing so it is possible to know how the system will behave in the presence of faults in its components or in its environment. Fault injection can be used for studying the effects of software or hardware faults. However, in both the academic and industry, most fault injection studies have aimed at the effects of physical hardware faults. Fault injection approaches vary according to the system's life cycle where they are applied, and the type of faults that are injected. Among the various existing approaches [4], Software fault injection is getting more popular.

Our proposed modified fault injection model includes nine elements such as UIF(User interface fault), MF(Memory fault), CF(CPU Faults), FSF(File system faults), APIF(API faults), NF(Network faults), RIF(Register information faults), IF(Initialization faults), DF(Disk system faults).According to the Proposed modified fault injection model, these nine aspects will be injected in to tested component.

The testing approach for tested component according to the modified fault injection model is shown as figure 1.Given a tested component TC, Modified model of testing approach will produce the testing driver TD according to TC and DM.Various types of faults will be injected in to the wrapped component using the testing driver according to the new fault injection model. If the tested component will passes through the effective evaluation criteria then security vulnerabilities is not exist in the tested component.

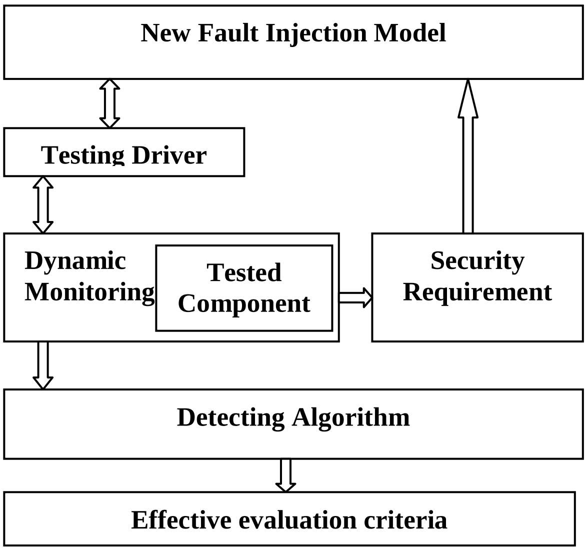


Figure 1.Testing approach of component security based on new fault injection model

III New Fault Injection Model

In recent years researchers and practioners have taken more interest in developing software implemented and physically interrupt injection tools. Software fault injection techniques are attractive because they don't require expensive hardware. Furthermore, they can be used to target applications and operating systems, which is difficult to do with the hardware fault injection.

Types of users generally include visual user (the human being) and the invisible users such as API,OS etc.[5].Based on the theory of invisible users and on theory of various types of faults, we propose new fault injection model, shown as figure 2.

Fault Injection Strategy and Environment

Based on this new fault injection strategy, faults will be injected in to the tested component which may include user interface faults, memory faults, CPU faults, File system faults, API faults, Network communication faults, Register information faults, Initialization faults and Disk system faults.

- Interface fault may include Input long parameter, Input incorrect parameter types, Incorrect parameter direction.
- Memory faults may include memory insufficiency, Segment locked, invalid address, excessive memory allocation.
- CPU Faults may include computation, control flow, and register faults.
- File system faults may include the restriction of reading and writing files, invalid files, corrupt directory and write protect-violation, converter required for reading the file etc.
- Process faults may include resource insufficiency, invalid type etc.
- Network communication faults may network congestion, invalid port, nonexistent or incorrect route setup, login incorrect and network unreachable.
- Register information faults Access denied, corrupt registry, unfound key.
- Initialization faults may include uninitialized or wrongly initialized variable or parameter.
- Disk system faults may include input/output error etc.

In origin, fault injection techniques were able to emulate hardware faults by physically interfering with the target system through special and expensive devices. To overcome limitation of such techniques, software implemented fault injection technique were proposed. They inject hardware faults by emulating the effects of faults (e.g., CPU or memory faults), i.e. corrupting the state of software using bit flipping or stuck at techniques. These techniques are also known as error injection.

Based on the hardware and software faults, we adopted the new fault injection model shown in figure 2.NFIM shows the above mentioned 9 aspects injected into the tested component. According to the different requirement in various application and test objects, NFIM can be extended. This strategy is having the injection of various types of faults which may include the software faults, hardware faults, design faults etc.

The disk system faults are injected by executing a routine in the driver code that emulates the I/O error. The injection of interface errors is another form of error injection where the error is specifically injected at the interface between modules (e.g. System components or functional units of a program).

A Fault injection environment typically consists of the following components:

- Fault Injector: Injects faults in to the target system as it executes commands from the workload generator.
- Fault library: Stores different file types, fault locations, fault times and appropriate hardware semantics or software structure.
- Workload Generator: Generates the workload for the target system as input.
- Workload library: stores the sample workload for the target system.
- Controller: Controls the experiment.
- Monitor: Tracks the execution of commands and initiate data collection whenever necessary.
- Data Collector: Performs online data collection.
- Data analyzer: Performs data processing and analysis.

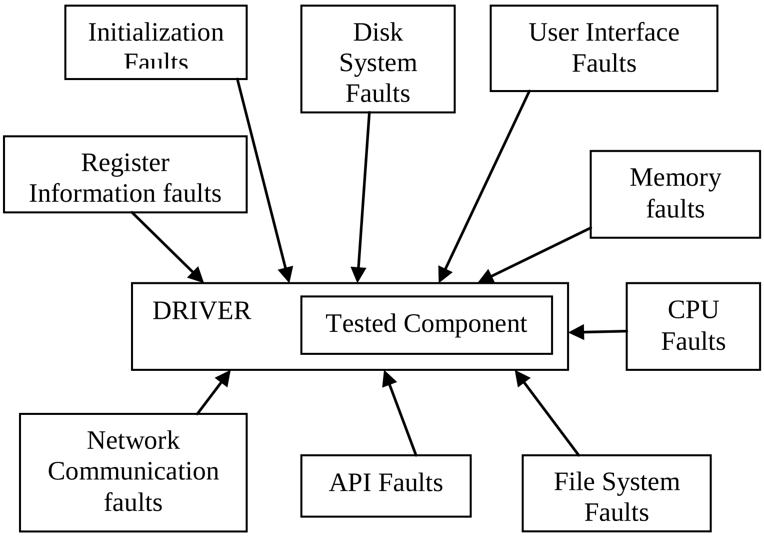


Figure 2 .Modified Fault Injection Model (MFIM)

IV Objectives of fault injection

Fault injection generally tries to determine whether the result of the system actually matches with the specifications, in presence of the various types of faults which may be the hardware faults, software faults, emulated based faults, simulation based faults. Generally various types of faults may be in the combination of two or three faults are injected in perfectly chosen system states and points. Tester knows the design in depth so it designs the test cases based on structural criteria.

There are several issues that can be considered in fault injection strategy such as what the faults inserted are, the combination of different faults type, the fault location, the faults frequency etc. The main focus of fault insertion is to identify all possible ways a software product can fail. Due to large amount of space involved in the software product, the fault injection process requires planning regarding the injection various types of faults.

V Software reliability metrics

Reliability metrics are units of measure for system reliability and are used for software reliability evaluation and assurance. Reliability metrics assess the degree to which a software product consistently performs its intended function without failure. Now in the direction of fault tolerance we are proposing a modified fault injection model with some new number of faults so when all these faults are injected in to the component then we will get some more accurate failure rate. Now on the basis of this failure rate, we are proposing a new metric for improving the measures of CBSE process. In general Mean time to failure (MTTF) is given by total operating time of units divided by total no of failures and our proposed metric will be like

$$\text{MTTF}=\frac{\text{Total Operating time of units}}{\text{Failure rate of component analyzed by modified fault injection Model}}$$

Therefore as the failure rate will get improved by MFIM then our mean time to failure metric will get improved.

VI Result

We have created a components repository in which there are fifty components available in repositories and these components are the application components, Microsoft IE components etc. These reusable components have written in different programming languages.

Now on the basis of our new testing technique, we have tested these components to find the reliable components. During the testing of components we are able to find out the cyclometric complexity of the component, DFD of component, Independent path of the component and also the test cases of components with the help of our new testing technique. As a result we will get the optimal number of components and with the help of these optimal number of components, the complexity of the software will get improved. The results showed that our approach was effective and operable.

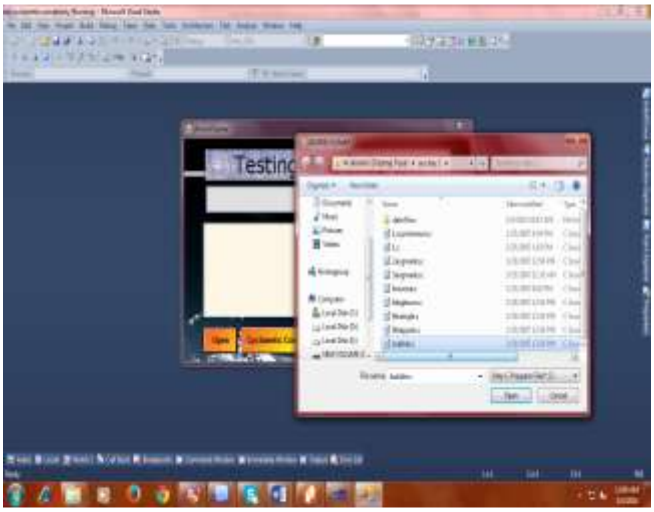


Figure 3. Insert the Component in Software

Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-073-6 doi: 10.15224/ 978-1-63248-073-6-25

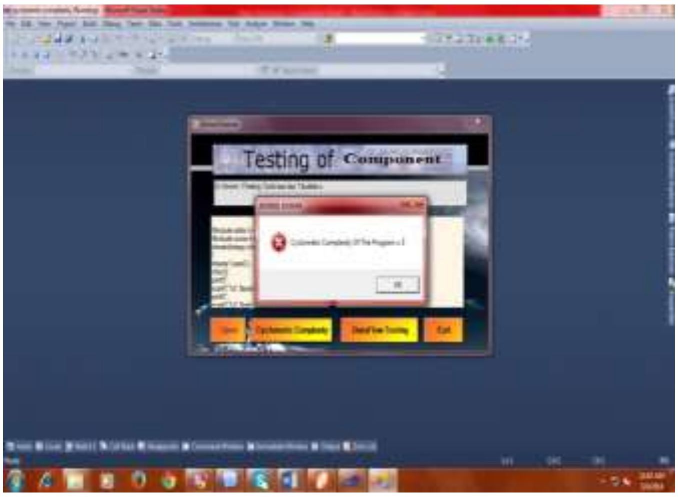


Figure 4. Showing the Cyclometric Complexity of Component

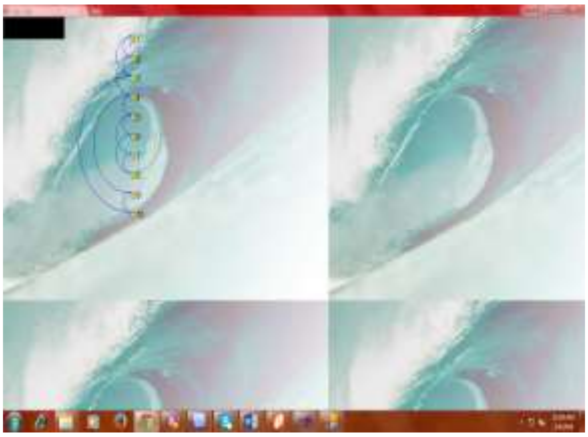
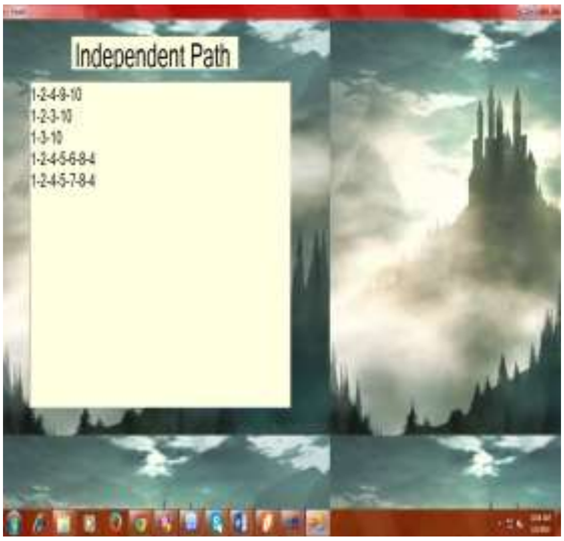


Figure 5. Showing the DFD of Component



Independent Path

1-2-4-9-10
1-2-3-10
1-3-10
1-2-4-5-6-8-4
1-2-4-5-7-8-4

Figure 6. Showing the Independent Path of Component

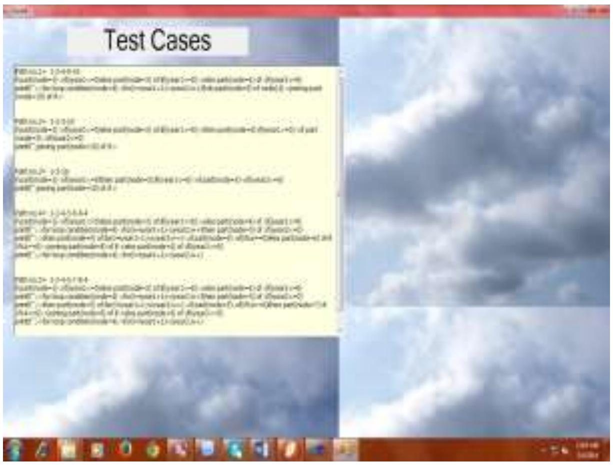


Figure 7. Showing the Test Cases of Component

VII Conclusions and Future Work

This paper proposes a testing approach of components in CBSE environment based on a new technique or new strategy and then defines and discusses the new modified fault injection model with the various new faults including the hardware and software and design based faults. Our results verified its integrity and operability.

As we know that the complexity of the software is directly related to the reliability of the system. So if the complexity of the components can be determined then it can be related to software reliability factor of CBSE environment and it's metric. There will be much important work in future research. A testing tool of automatization should be developed based on MFCVD approach and some fault detection technique can be related to some reliability metric and the metric can be improved and automatically complexity of the software system will be enhanced.

References

[1] J.Chen, Y. Lu, and X.Xie, "Testing Approach of component security based on fault injection", Huazhong university of science and technology, 2007.

[2] C.Jin-fu, L.Y.Sheng, Z.Wei and X.Xie-Dong "A fault injection model oriented testing strategy for component security" , Huazong university of science and Technology, 2009.

[3] J. Han and Y. Zheng, "Security Characterization and Integrity Assurance for Component Based Software" , *IEEE Software,PP.61-66*,2000

[4] Hsueh, M.C; Tsai T.;Iyer, "Fault Injection Technique and Tools" , IEEE Computer,April/1997

[5] J.AWhittaker, "Software Invisible Users, IEEE Software, vol. 18,*pp. 84-88,* june 2001.

[6] S.Yacob, B.Cukic, and H.Ammar, "A Scenario based reliability analysis approach for component based software" , IEEE Transactions on relaibility,Vol 53, December 2004.

[7] J.Gao,M.C.Shih. A Component Testability model for verification and measurement [C]// Proceedings of the 29th annual international computer software and application confrences(COMPSAC '05), San Jose,USA ,IEEE,2005:0730-3157/05.

J.Gao,D.Gopinathan,Q.Mai,J.He. A Systematic Regression testing method and tool for software components[C]// Proceedings of the 30th annual international computer software and applications confrences(COMPSAC '06), San Jose,USA ,IEEE,2006:07695-2655-1/06.