Volume 4, No. 3, March 2013

Journal of Global Research in Computer Science

ISSN-2229-371X

REVIEW ARTICLE

Available Online at www.jgrcs.info

COMPATIBILITY ESTIMATION FOR COMPONENT BASED SOFTWARE ENGINEERING

Dhawaleswar Rao. CH[1], Nishant Gupta[2]
[1]Assistant Professor, Department of Computer Science & Engineering,
Lovely Professional University, Jalandhar (Punjab), India,
dhawaleswarrao@gmail.com, dhawaleswar.14671@lpu.co.in
[2] M.Tech Student, Department of Computer Science & Engineering,
Lovely Professional University, Jalandhar (Punjab), India,
nishantdnn@gmail.com

Abstract: The Present time system software's are complex and cost estimation is required in such type of complex software's .The overall cost of such software's can be reduced by reusing the existing components. The component based software engineering is the technique through which we can easily reuse the existing components and able to reduce the overall software cost. We can estimate the overall cost of the system software before development, if the overall cost will be reduced by reusing the existing components then only we can reuse these components otherwise not. The Compatibility is the main problem in the component based software's. Before reusing the component based software module, we have to check its compatibility with the software in which we are using that module. In our work; we have designed an automation tool to check the compatibility. The overall system cost can also be increased when we perform integration testing. Our proposed new technique will reduce overall software cost and provide better integration testing.

Key Words: Component based, Cost estimation, compatibility

INTRODUCTION

In the past times the software's are developed originally without reusing the existing components. But now a day the software's are much complex and large .In such type software's we have to reuse the existing components to reduce the overall cost of the system software. The component based software's are those who software module follows the principle of plug and play. Software components are easy to reuse. Object oriented software components are the independent components that can be plugged in the plugged out in the software system. The plugged in and plugged out of the software component will not effect the exciting software system. These types of reusable components are easy to reuse.

The cost estimation can be done properly in component based software. We can only reuse the existing components if the overall cost of the system software can be reduced. The developers need to put extra efforts on the developing the software components which will be reused in future. Object oriented software components have higher level of abstraction. These higher level abstractions object oriented components can be easily plugged in and plugged out without effect the exciting software system. When the components based reusable module is reused with the other system software, compatibility problem may exits. Let us suppose we are developing a new web based application, in this application we are reusing the login module of other we based application, if login module will not be compatible with the application compatible problem exits. To solve this problem before reusing the components we have to check its compatibility. In this work we are developing a new automation tool to check compatibility.

In this paper, Literature Review will be presented in section 2. Component based approach will be written in section 3.New proposed technique will be explained in section 4.Tool and simulation results will be shown in section 4.In the last section 5 future work and conclusion will be presented .

They proposed a new approach which consider the severity of fault based on requirement prioritization .Main aim is to find the severe fault early in the testing process and hence to improve the quality of the software according to the customer point of view [1].

They had presented BCO algorithm for maximum fault coverage using two examples whose results are comparable to optimal solution .In this they using Average Percentage fault detection metrics and charts have been used to show the effectiveness of the proposed algorithms [2].

They had discussed about the direct relationship between the attributes and the quality factors. They classified attributes as external product attributes and internal product attributes. The external product attributes are those which influence the external behavior of the system had how system behaviors in the external environment. The internal product attributes are those which influence the size, cost and performance of the system [3].

They had discussed the object oriented technology of software re-engineering and define objects, attributes, values. They had discussed that object oriented components are independent components which can be pulled in and pulled out according to the need in the software system [4].

In this paper they had discussed about the need of re-engineering and with the process we can reduce the software cost and time. To effectively implement the re-engineering we need certain set to parameter metrics .In the cost of the software decreases by using the re-engineering matrices then only the re-engineering can by done otherwise we adopt the tradition mechanism of software development that is fresh development [5].

In this paper they had described the number of regression test selection techniques for different programming paradigms such as procedural, object-oriented, component-based, database, aspect and web applications. In this paper they review the important regression test selection techniques proposed for various categories of programs and identify the emerging trends [6]

They had proposed Hybrid technique based on BCO for analyzing text case selection. Their result shows that a huge amount of reduction in test suit takes place. Reduction in test suit reduces time as well as cost. They have proposed hybrid approach which proves much faster than ACO technique. The tool which they developed runs much faster to provide the minimum subset of test cases .The tool can provide different result in each group. This approach will reduce the time for testing the software and we can easily identify the faulty objects in the object oriented software system [7].

COMPONENT BASED SOFTWARE ENGINEERING

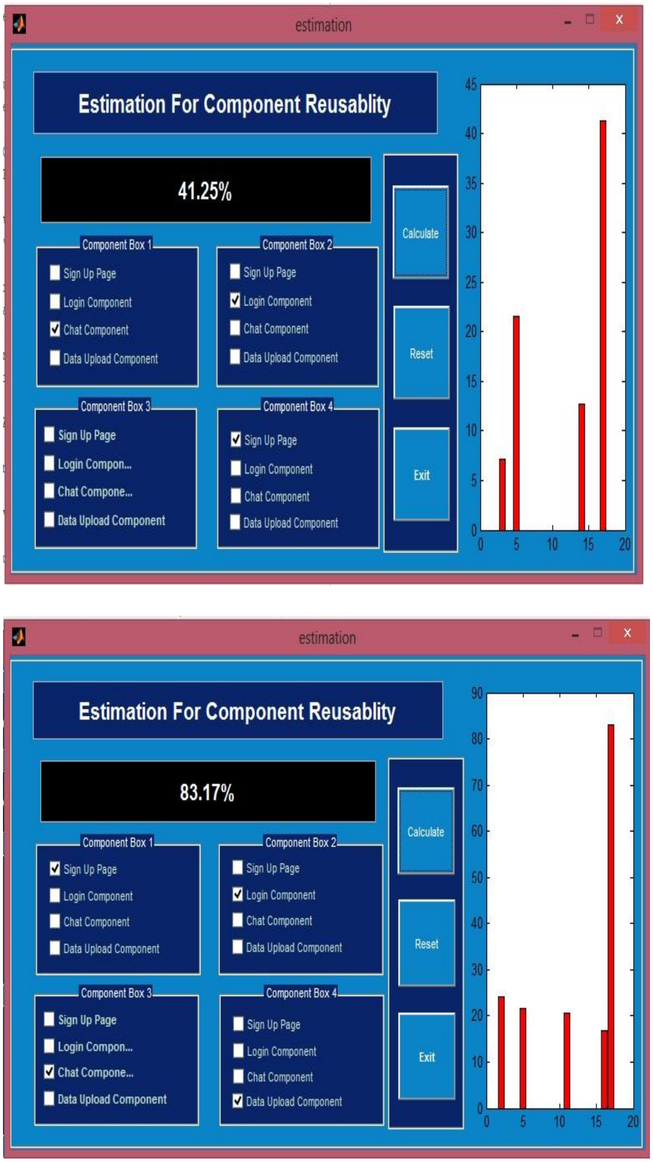
The object oriented software's in which the modules can work independently and each module can be pulled in and pulled out from the software system when required. Designing an object oriented software system is very hard but designing a reusable object oriented software system is even harder. When designing a reusable object oriented software components the design should be able to solve the future problems and must compatible with the most software systems. While designing a system software we follow the tradition approach of fresh development and if we want to reduce the cost and time of development, we use the object oriented reusable components. To estimate the cost of system software, certain set of metrics are used. While, developing system software our main motive is to reduce the cost and time of devolvement, if overall cost of the system software is reduced by using object oriented software components then only reusable components are used otherwise not. The object oriented components when pulled in the system software there will be a chance that they will gave incorrect responses to certain set of inputs. The incorrect responses can be due to the faulty objects that excites in the reusable components. To identify these faulty objects in the object oriented software components, we need reengineering approach. The reengineering approach is very time consuming and inefficient, it is very hard to identify these faulty in the reusable software components, in the complex software systems it even harder to identify these faults objects.

NEW PROPOSED TECHNIQUE

The new technique is proposed in this paper to identify the number of faulty objects in the component based system software's. The objects which create the compatibility problem in the system software's are referred as faulty objects. In this work, we had developed an automation tool .The automation tool will identify the compatibity percentage with the new system software .The modules which having higher compatibity with the system software are included in the system software.

Automation Tool:

In figure 1, the automation tools which are shown. This is based on the certain component based system software modules which is designed for different software's. In display window the calculated value is shown and bar graph is shown in the tool.



estimation

Estimation For Component Reusablity

41.25%

Component Box 1
☐ Sign Up Page
☐ Login Component
☑ Chat Component
☐ Data Upload Component

Component Box 2
☐ Sign Up Page
☑ Login Component
☐ Chat Component
☐ Data Upload Component

Component Box 3
☐ Sign Up Page
☐ Login Compon...
☐ Chat Compone...
☐ Data Upload Component

Component Box 4
☑ Sign Up Page
☐ Login Component
☐ Chat Component
☐ Data Upload Component

estimation

Estimation For Component Reusablity

83.17%

Component Box 1
☑ Sign Up Page
☐ Login Component
☐ Chat Component
☐ Data Upload Component

Component Box 2
☐ Sign Up Page
☑ Login Component
☐ Chat Component
☐ Data Upload Component

Component Box 3
☐ Sign Up Page
☐ Login Compon...
☑ Chat Compone...
☐ Data Upload Component

Component Box 4
☐ Sign Up Page
☐ Login Component
☐ Chat Component
☑ Data Upload Component

Figure 1: Automation Tool

In the above figure the working of the automation tool is shown with selecting the different parameters. The outputs are shown in the form of percentage and graphs

CONCLUSION AND FUTURE WORK

In this paper we conclude that the component based software engineering is the efficient approach for dealing with the higher complex and real time software systems. The system software cost will be reduced by reusing the component based software modules. The compatibity problem may exits in this type of system software due to the faulty object. To check the compatibility of these types of component based software modules the automation tool is developed. In future, we will work on to develop such a compiler based tool which will identify the faulty objects in the component based system software's.

REFERENCES

[1]. Dr. Varun Kumar, Sujata, Mohit Kumar "Test Case prioritization using faulty Severity", IJCST International Journal of computer Science and Technology, Vol. 1, Issue 1, September 2010

[2]. Dr. Arvind Kaur, Shiavngi Goyal "A Bee Colony Algorithm for Fault Coverage based Regression Test Suit Prioritization" IJAST, Vol.29, April 2011

[3]. Phuc V, Nguyen, Ph.D candidate "The Study and approach of software reengineering." IJAST, Vol.29, 2010

[4]. Dr.Ashok Kumar, Bakhshsish Singh Gill "Cost Model for Reengineering an Object Oriented Software System" Global Journal of Computer Science and Technology Volume 11 Issue 20 Version 1.0 December 2011

[5]. DR.LINDA, H.Rosenbergy "Software Re-engineering"

[6]. Swarnendu Biswas and Rajib Mall,"Regression Test Selection Technique: A Survey, Informatics 35 (2011)

[7]. Bharti Suri and Isha Mangal," Analyzing Test Case Selection using Proposed Hybrid Technique based on BCO and Genetic Algorithm and a Comparison with ACO," Volume 2, Issue, April 2012

[8]. Robert S. Arnold, "Software Reengineering", IEEE Computer Society Press Los Alamitos, California