Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 85 (2016) 922 – 928

Procedia Computer Science

International Conference on Computational Modeling and Security (CMS 2016)

A Design Based New Reusable Software Process Model For Component Based Development Environment

Ms. Jyoti sharma[a] , Dr. Arvind Kumar[b], Ms. Kavita[a,b,*]

[a]Ph. D [CSE] Research Scholar, SRM University, Delhi-NCR Sonepat, Haryana-131029, India
[b]Department of Computer Science & Engineering, SRM University, Delhi-NCR Sonepat, Haryana-131029, India

Abstract

Software development considered to be an important part of software industry. Various metrics, algorithms and reusable process models has been designed but ultimately our main goal is only to find that part which will help us to select the optimal one which may be a metric, algorithm or a reusable software process model. For the various large applications some components need to be built separately and some of the components need to be modified according to the requirement for searching the optimal components. Now a day's component based software engineering considered to be the best approach for the software development at low cost and this software development best approach will totally dependent on the optimal selection of components. The aim of this paper is to describe the characteristics of some selected state of art CBSD models and a new reusable software process model has been designed for the optimal selection of components based on the new optimal algorithm.

Keywords: *Component complexity;coupling;cohesio;reusable model*

1. Introduction

Component based development offers a radically new approach to design, construction, implementation, and evolution of software applications. Software applications are assembled from components from a variety of sources; the components may be written in several different programming languages and run on several different platforms or we can say that components are heterogeneous in nature.

To ensure the delivery of quality component based software, effective optimal selection is necessary for the software development. Today complex, high quality component based system can be built in very short period of time by using the effective component selection technique. Modern software systems become more and more large scale, complex and uneasily controlled, resulting in high development cost, low productivity, unmanageable software

* Dr. Arvind Kumar   Tel.:+91 8295032966
E-mail address: k.arvind33@gmail.com

Peer-review under responsibility of the Organizing Committee of CMS 2016
doi:10.1016/j.procs.2016.05.283

quality and high risk to move to new technology [4].Generally the process for the component based software development comprises of two separate processes: (1) for component development and secondly for system development and during both the processes optimal selection of components will play an important role. The standard development process for the component based systems does not include the optimal selection of components .This paper studies the new reusable software process model that will helps in the optimal selection of components for the system development.

2. Literature Survey

Various CBSD models have been designed; some of them are studied and described briefly:

2.1 *The X Model*

The X model is named as X because its shape has been given as in X letter. In X model the process starts by requirements engineering and requirement specification as shown in figure 2.This X model is basically consisting of development for reuse, development after modification, component based software development, development without modification.

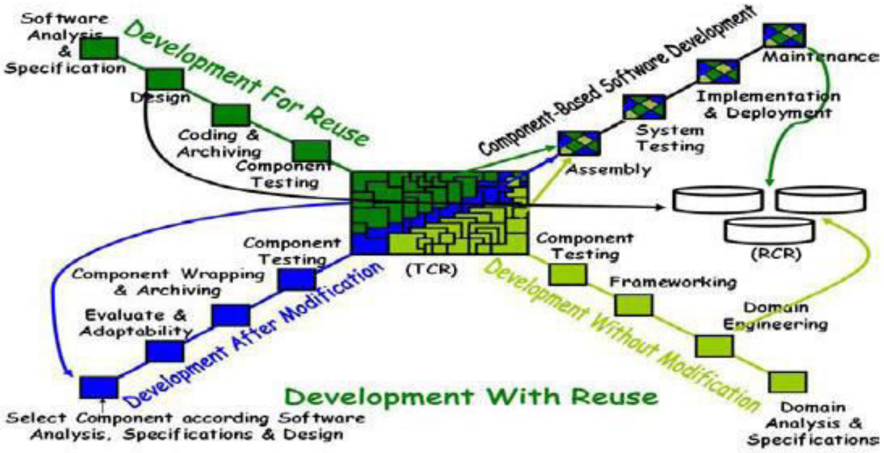

Software Analysis & Specification
Development For Reuse
Design
Coding & Archiving
Component Testing
Component-Based Software Development
Maintenance
Implementation & Deployment
System Testing
Assembly
Component Testing
(TCR)
Component Testing
(RCR)
Frameworking
Domain Engineering
Domain Analysis & Specifications
Development Without Modification
Development With Reuse
Component Wrapping & Archiving
Evaluate & Adaptability
Development After Modification
Select Component according Software Analysis, Specifications & Design

Fig 1: The X model

Two approaches can be used to select/find the component, first is software development with modification in reusable component and second is development without modification in reusable component. Component can be selected from two repositories Testable Component Repository (TCR) and Reusable Component repository (RCP).

X- Model is best for large software development but it mainly focus on reusability not on risk analysis and feedback. The complexity of this model increases the cost too and due to the increase in cost this model cannot be selected for selection of optimal components from the component repository.

2.2 *The Y-Model*

Y- Model was proposed by Carpretz for component based development environment.

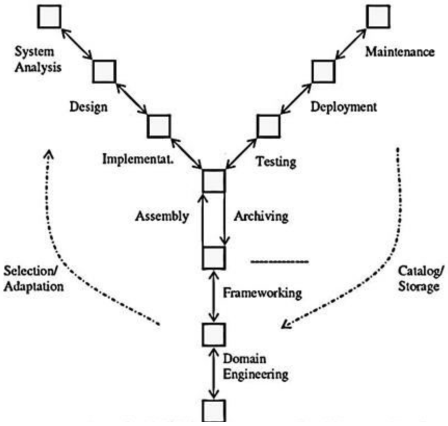


Fig 2: The Y-Model

This model includes some basic activities with some additional activities as follows domain engineering, frameworking, assembling, archiving, system analysis, design, implementation, testing, deployment and maintenance. This model follows both top-down and bottom-up approach. Iteration and overlapping are becomes the recede of this model.

2.3 *The V-Model*

V model is adaptation of the traditional waterfall model for building a system from reusable software components.

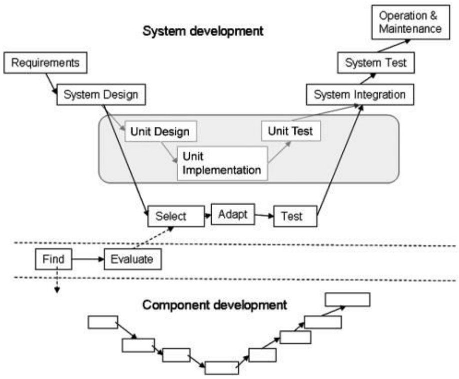


Fig 3: The V-Model

It consists of several steps that include requirements, system design, unit testing, system integration, system testing, operation and maintenance. Testing is main concern in this model; in the end of every phase testing gets done.

2.4 *The Knot -Model*

In knot- model the uttermost emphasis is given on reusability, risk analysis and feedback in each and every phase, that results a simple and error free system. The selection of component phase gives three possibilities:
1) Use existing component as it is.
2) Use existing component with modification
3) Develop the component from scratch.
This model reduces the complexity, risk, development time and cost.

3. Optimal Selection of Components with Proposed Reusable Model

Optimal selection of components by using the reusable model will use the following number of parameters such as NDIUC, DUM, LC2P and HC2P.

NDIUC:-It is the number of direct or indirect used classes or can be said as the direct or indirect interaction. Direct interaction means the direct use of the method of a class with in a package and Indirect interaction means the use of a method by importing the particular package.

An ideal system with the hierarchical structure can be shown as:-

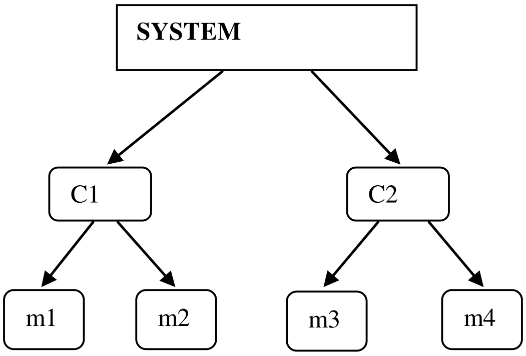


Fig 4: Ideal hierarchical system

Where C refers to as the components and m refers to as the methods. The general hierarchy which is generally followed in java source code as:-

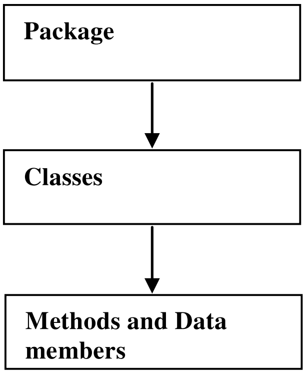
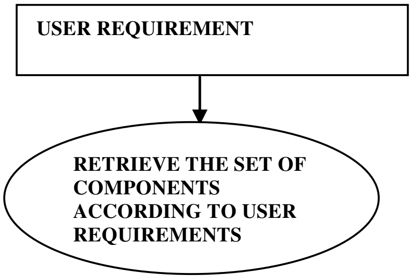


Fig 5: Hierarchy of packages

DUM refers to the set of all direct connection between classes and methods
LC2P refers to the low complexity cohesion package.
IUM is the set of all indirect connection between classes and methods.
HC2P will be the high complexity cohesion package which is measured in terms of direct used classes and indirect used classes.

3.1 Proposed Reusable component selection model


USER REQUIREMENT
RETRIEVE THE SET OF COMPONENTS ACCORDING TO USER REQUIREMENTS

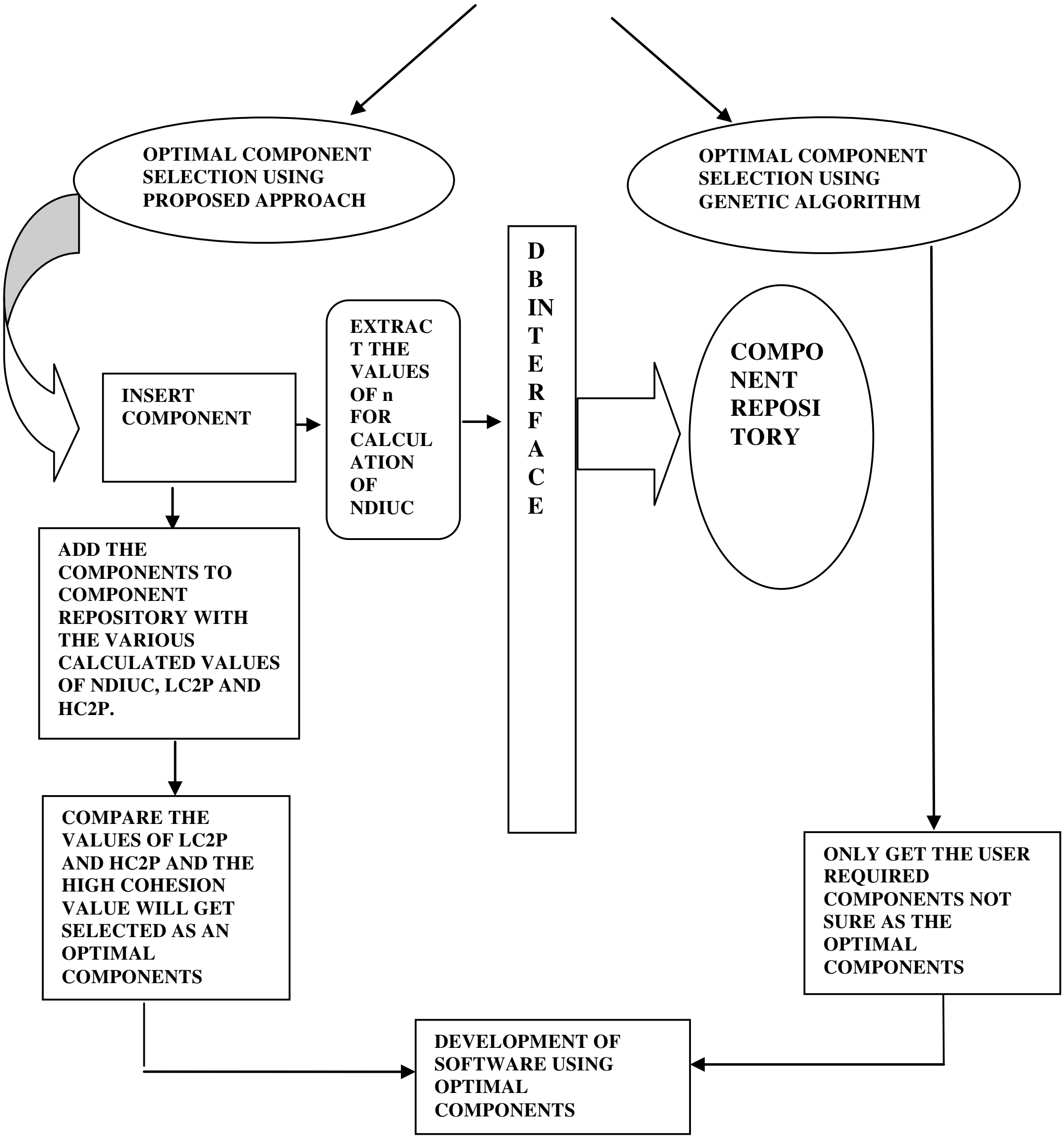

OPTIMAL COMPONENT SELECTION USING PROPOSED APPROACH
OPTIMAL COMPONENT SELECTION USING GENETIC ALGORITHM
D B INTERFACE
INSERT COMPONENT
EXTRACT THE VALUES OF n FOR CALCULATION OF NDIUC
COMPONENT REPOSITORY
ADD THE COMPONENTS TO COMPONENT REPOSITORY WITH THE VARIOUS CALCULATED VALUES OF NDIUC, LC2P AND HC2P.
COMPARE THE VALUES OF LC2P AND HC2P AND THE HIGH COHESION VALUE WILL GET SELECTED AS AN OPTIMAL COMPONENTS
ONLY GET THE USER REQUIRED COMPONENTS NOT SURE AS THE OPTIMAL COMPONENTS
DEVELOPMENT OF SOFTWARE USING OPTIMAL COMPONENTS

Fig 6: Reusable Component Selection Model

3.2 *Selection of Components*:-

Generally the user requirements are there to predict the type of model requirement. When the user requirements are given then we will find the optimal number of components using the proposed reusable software process model is as follows:-

1) Gathering of components which may be in any type of source code.
2) Coupling and cohesion parameters are calculated as:-
   NDIUC = m (m-1)/2

LC2P = m (DUM)/NDIUC

HC2P = m [DUM U IUM]/NDIUC

3) Apply the reusable process model.
4) Compare the values of LC2P and HC2P.
5) Select the components with the good values of HC2P so that these components considered being as the optimal components.

Actually whenever will get the user requirement then we have to select the optimal components by using the proposed reusable model approach or by using the existing genetic algorithm. The main difference between these two is by using genetic algorithm approach we will get the components according to the user requirement but not sure as the optimal components one but by using our proposed approach we are able to select the optimal number of components according to the user requirements so it can be considered to be as the best.

4. Conclusions and Future Work

This paper proposes the new reusable software process model for the optimal selection of software components in CBSE environment based on a new strategy. During the optimal selection of components if the selection is done by using the genetic approach then it will helps us to select the quality components but not sure as the optimal components for the development of software But by using the proposed approach the optimal selection of components can be done.Our results verified its integrity and operability. Some of the features of the reusable software process model are:-

1. This model can be used to find the optimal components for the development of component based software.
2. This model can be used to evaluate the quality and efficiency of components. The LC2P and HC2P values calculated above helps to estimate the complexity of the software.

In this paper, we proposed a reusable software process model for the component based software by using potentially important parameters. The proposed approach can be applied to other projects also.

5. Comparison With Previous Approaches

Various hybrid approaches which were two phase approaches has been designed and also the keyword based search approaches were there. Recently genetic algorithm was used for selecting the user required components where the complete algorithm was based on the termination condition of genetic algorithm. This dynamic approach for the retrieval of software components using genetic algorithm was used to retrieve the best user required components not the optimal components.

The proposed reusable model will work to find the optimal number of components from the component repository according to the user requirement. This reusable model uses the various parameters. The components with the high value of HC2P will be selected as the optimal component. The effect on various parameters will be as follows:-

A. PERFORMANCE:-

If we are capable of finding the optimal number of components then software development process will get automatically improved therefore there will be a direct impact on performance of the whole system.

B. COMPLEXITY:-

Optimal selection of components will automatically improve reliability of the system and because complexity is directly related to the reliability of the system so complexity will get automatically improved.

C. Time

By using this proposed approach if the optimal selection of components can be performed then software development will get easier and side by side the time for development will get automatically reduced.

References

1. W.Stevens, G.Myers, and L. Constantine.Structured design ,IBM Systems Journal, 13(2):115-139, 1974.
2. T.J McCabe,A Complexity measure,IEEE Transactions on software Engineering,SE-2(4),Dcember 1976, 308-320.
3. L.Yu, Kai.C and Srini.R, "Multiple Parameter coupling metrics for layered component based software" ,University of Arkanas at little rock,USA,2008.
4. G.Pour, "Software Component Technologies:JavaBeans and ActiveX,"Proceedings of Technology of Object Oriented Languages and systems,1999,pp 398-398
5. A.Yadav and RA.Khan, "Class Cohesion Complexity Metric" ,Babasaheb Bhimrao Ambedkar University,ICCCT,2011
6. J.AWhittaker, "Software Invisible Users, IEEE Software, vol. 18,*pp. 84-88*, june 2001.
7. E Da-wei, The software complexity model and metrics for object oriented. School of computer Engineering, Jimei University,Xiamen, China, 16-18 April 2007,464-469.
8. 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.
9. Capers Jones, Applied software measurement: Global Analysis of productivity and quality, McGraw-Hill, New York,2008.
10. Hu Xiuwen, Tian Zhonghe, Research on Software Metrics[J].Information Technology, 2003, 27(5) :58-60

S.Mahmood and R. Lai, "A complexity measure for UML component-based system specification."Software-Practice and Experience,John Wiley and sons,2008, 38(2),117-134.