Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.8, No.1, 2017

An Effective Approach for Evaluation and Selection of Component

Faisal Islam
Department of Computer Science, University of Agriculture, Faisalabad, Pakistan

Abstract
The main objective of the research is to find the best optimal component and define the evaluation criteria for the reusable software component from the component repository that increases the quality of software component, reliability and also reduces the software development cost. In current scenario of the software development the main factors like time and cost for the component selection is too short and is also very difficult to identify the best matching component from the component repository. Various research and methodologies for the best optimal component selection models are exist with relevant to the data validation and verification to translate the annotated models into component performance. The decisions are often taken on temporary basis for the component selection from the component repository. The pre existing methodologies and processes for the component evaluation and selection does not meet the characteristics and evaluation of the functional and non-functional requirements. An effective solution for the evaluation and selection of the reusable software component from the component repository is a very hot issue for the software development community. In this paper the research will concentrate on the issue that belongs to the evaluation and selection process to find the best optimal component from the component repository by using the genetic algorithms.
Keywords: Components, selection, Reusability, Genetic algorithm, Component Retrieval

1. Introduction

Component based software engineering (CBSE) is bases on the evaluating, selecting, composing, designing various software components. To select the best possible component from the component repository is challenging task that addresses the customer requirement with minimum development time and cost. This research will addresses issue that are relevant to the evaluation criteria and selection process for the component during manufacturing of the component based software systems. The main issues in such component based systems which software component is selected and which type of verification needed and which kind of pre existing software components in the component database (Geisterfer & Ghosh 2006).

At the early of the software development process the decision for the evaluation and selection of the best optimal component must be carried out. The requirement of different projects are directly associated with the foremost projects such as work flow of organization, resource management, planning coordination between stakeholders. These wells organize and associate projects play an important role in component based software development that likes a catalyst. For this purpose does not rely on single methodology, try to uses some best low cost alternative component instead of expensive ones. The main advantage of component based software engineering is to get parallel and distributed development as well as increase reusability at lower cost with increasing flexibility. In previous research many different models are used in component based software engineering such as V model, Spiral model, CMMI and ISO 9000 all these models are used to addresses a particular projects. Object Oriented Development OOD that describes the methodology and processes like RUP (Rational Unified Process). All these methodologies and processes are still focused on the designing, system specification and implementation (Crnkovic *et al.* 2006).

Lot of principles of component based development which influences the development processes and maintenance but these required considerable modification to attain standard values. The most important benefits of this approach is to decrease the time to market, reduced the development time and cost, so for the software engineer and developer select the highly specified components on need bases and then assembles these selected component into reliable system (Fonoage *et al.* 2010).

2. Previous Research Work

Kwong *et al.* (2010) described that the first stage in the component based software engineering is to select the software components from the component repository. After finding the best software components the next stage is to select the best optimal component for each software components. This optimization model has two major phases, in the first phase the functional performance of the component based software system increased and in the second phase the cohesion functionality is increased but decreased the coupling functionality. The conventional development approaches is not suitable for developing a large and complex software systems using the reusable software components because these approaches try to develop a software system using scratches. The author described the new perspective which is called synthesis. The technique provides the software development environment in which try to decrease software development time, cost and human effort for

independently software development. All these independent software components are assembled together to developed new large and complex systems.

Kaur & Goel (2011) described the major benefits of the component repository that store a large number of component in the organized and systematic manner. Component repositories should be capable to addresses the changing and growing of the software organization and industries. To the retrieve the best optimal component from the large component repository is the most important element of the reusable software components. The author also described the different aspects of the component repositories such as component searching methodologies and classification like Free Text, Enumerated, Attribute value and faceted classifications. The end users and software developers can design customized and high level queries to retrieve the effective component from the component repository according to the requirement of the end users.

Indumati *et al.* (2012) described that the software developer and organization facing lot of problems of an effective very large scale software applications. Especially knowledge based software is very complex in nature and very large in size which is also very difficult to handle for the software developers and organizations. There are two major tasks that produce good quality software, the number one is software complexity and the second number is software quality measurement. To handle such situation modular software applications presents a better solution for very large and complex software systems and also provides their functional requirements with respect to size. Different software components are integrated and assembled with each other and these components make a large application and modules. The basic problem in the component based software development is to select the best optimal component from the component repository which produces good quality software at low cost and too short time.

Tomar & Gill (2013) proposed a new algorithm that has two major strategies for the selection of best optimal component from the component repository; the first strategy belongs to best fit and second is first fit strategy. In the proposed algorithm applications is developed using the X model with two approaches of the component based software development. The first approach is development with reuse and the second is development for reuse. The development with reuse approach is used for the already developed software components. The selection of the best optimal component from the component repository is a very attractive and challenging job for the software developers. In this researcher also described two main issues for the component selection. The first is Simple Component Selection Problems (SCSP) and second is Criteria Component Selection Problem (CCSP). To select the best optimal component from the component repositories with the help of x model to satisfy the customer requirements by using the Simple Component Selection Problems (SCSP) and Criteria Component Selection Problem (CCSP). Component based software engineering is an effective approach to increase the reusability of the component based software development from the pre existing components or with the development from scratches.

3. Component Selection Model

The most important advantage of software reuse is to reduce the development cost and time for the retrieval of best optimal component from the component repository. The development of the keyword based software for the retrieval of the best component from the repository is very easier and easy to modify than the classes systems. The information retrieval system is divided into steps from the component repository the best possible solution of the problem is to be found. The information that is retrieved from the component repository is often very complex and complicated because no one knows which is the best solution for the system that is satisfy the customer requirements. After completing the first stage the process move to the next stage that is the selection of the optimal component by applying the various operation of the genetic algorithms. The most important role of the genetic algorithms is to find the best possible solution for the application according to customer requirements.

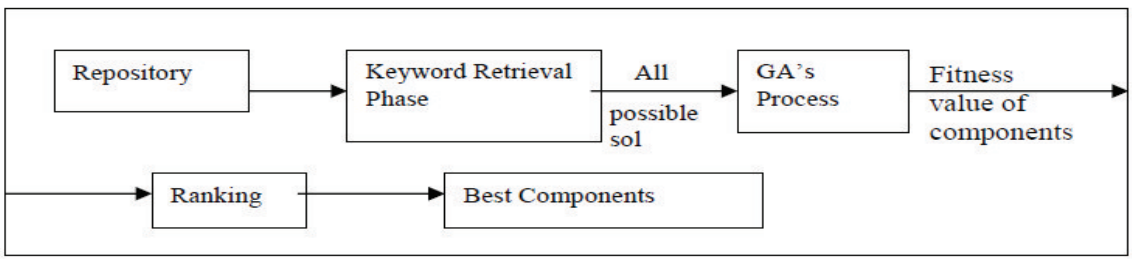

Figure 1: Abstract Diagram of Component Selection Model

These genetic operators are applied on a large number of populations and repeat the process until specified condition is fulfilled. After the completion of the genetic operators' procedure the next step is to the ranked the component against their fitness values. Retrieval process provides the best possible components or optimal solution.

3.1 Component Description

The component repository is also known as component database that stores various components. Component based software engineering requires a large repository for component retrieval that becomes important to supply components. Component description is categories into three stages.

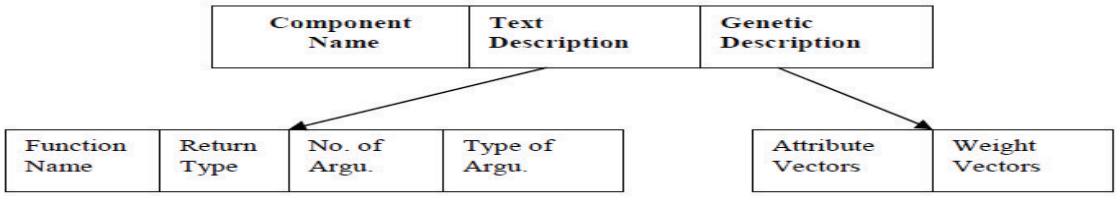

Figure 2: Component description

3.2 Component Name: It is the name in repository that has the unique identification.

3.3 Text Description: It includes functionality of the component; return type of the function, function used arguments, arguments type

3.4 Genetic Description: If attribute present in the component then assign value "1" to the attribute otherwise assign value "0" to the attribute. The attribute vector uses binary scheme of coding. The weights against the attribute vectors are also stored in the genetic description. The weight vector gives the most important rank to those attribute whose aggregate value in the component are high. The weight vector value ranges from 0 to 1.

4. Optimization using Genetic Algorithms

In biological system that is depends on the mechanical phenomenon of naturally selected and natural genetics in the genetic algorithms. Genetic algorithms are vigorous in various field of application and sort out a complex problem to find a new solution with the improved performance while exploiting the information. The main objective of the proposed system is to evaluate and select the optimal component from the component repository using genetic algorithms (Sandhu *et al.* 2008).

5. Genetic Operations

Genetic operations include following stages.

5.1 Selection and Reproduction: By combining the highly fit chromosomes the genetic algorithms reproduction operator produce a fit individual. Both the processes are randomized the function of the first random process is to select the parent chromosomes and the function of the second random process is the steps inside the reproduction operator (Radwan *et al.* 2008). The selection of the chromosomes depends on the objective function. For which chromosomes are selected for reproduction that have the highest objective function of a chromosomes. In the reproduction process each selected parent chromosome is participated in the process.

5.1 Crossover

A single Site Crossover proceeds in three steps. First is the selection of two individual string pairs at random for mating in reproduction operator. Second is the selection of cross site is chosen at random and third is the String values are exchanged among two strings (Vrajitoru & Dana, 1998).

5.2 Mutation

Change bits incorporate paired qualities 0 and 1 and trade these qualities with little transformation likelihood Pm. The mutation probability is the probability in which number of bits is calculated for mutation. It is also very important in mutation search that the diversity between the populations is maintained. The aggregate number of bits changed in the population is relying upon the three parameter, transformation likelihood and population size and population length of the people in the population.

Number of changed bits = Probability mutation * Population size * String lengths (1)

5.3 Termination Criteria

For stopping the process termination criteria in genetic algorithms is very necessary. Genetic processes consist of large number of iterations. A termination criterion gives the important information when to stop the genetic process. After the predetermined number of period genetic procedure end the genetic calculations and test the personality of the finest individual from the population beside the characterized issue. There are two parts of the termination criteria

Fitness Convergence: An end approach that stops the advancement when the fitness worth is considered as joined. The distinction between the present population and the past population is not exactly the predetermined worth, the fitness quality is considered as merged. In this process the value 0.01 is used for the termination criteria. The calculated average fitness value is as (Fan *et al.* 1999).

$$\sum_{j=1}^{8} AjWj \quad (2)$$

AvrageFitness current pop – AverageFitness previous pop<= |0.01| the genetic process will stop if the specified condition is satisfied.

Generation Number: if the fitness convergence does not provide the required result then we use the generation number method. In this technique that stops the advancement when the predefined greatest number of development have been run.

6. Experimental Results

In the termination criteria the average fitness value of the first population is different from the average fitness value of the initial population. The difference between them is more than particular value in the termination criteria; it means that process is continuing.

Table 1: New population

Component Name	A1	A2	A3	A4	A5	A6	A7	A8	W1	W2	W3	W4	W5	W6	W7	W8	Fitness Value
C1	1	0	1	0	0	1	0	1	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	2.7
C2	0	1	0	0	0	0	1	0	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	1.3
C3	1	1	1	1	0	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C5	1	0	0	0	1	1	0	1	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C6	1	1	0	1	1	0	1	0	0.8	0.5	0.5	0.5	0.8	0.6	0.9	0.5	3.2
C8	1	0	0	1	0	1	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C9	1	0	1	0	0	1	0	1	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	2.7
C10	0	1	0	0	0	0	1	0	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	1.3
C12	1	1	1	1	0	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C13	1	0	0	0	1	1	0	1	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C14	1	1	0	1	1	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C17	1	0	0	1	0	1	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C18	1	0	1	0	0	1	0	1	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	2.7
C19	0	1	0	0	0	0	1	0	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	1.3
C20	1	1	1	1	0	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C22	1	0	0	0	1	1	0	1	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C24	1	1	0	1	1	0	1	0	0.8	0.6	0.5	0.5	0.8	0.6	0.9	0.5	3.3
C25	1	0	0	1	0	1	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C26	1	0	1	0	0	1	0	1	0.9	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C27	0	1	0	0	0	0	1	0	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	1.3
C28	1	1	1	1	0	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C29	1	0	0	0	1	1	0	1	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C31	1	1	0	1	1	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C34	1	0	0	1	0	1	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C39	1	0	1	0	0	1	0	1	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	2.7
C42	0	1	0	0	0	0	1	0	0.9	0.6	0.5	0.5	0.8	0.6	0.9	0.5	1.3
C45	1	1	1	1	0	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C46	1	0	0	0	1	1	0	1	0.8	0.6	0.5	0.5	0.8	0.6	0.9	0.5	2.6
C48	1	1	0	1	1	0	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.9
C51	1	0	0	1	0	1	1	0	0.8	0.1	0.5	0.5	0.8	0.6	0.9	0.5	2.4
C52	1	0	0	0	1	1	0	1	0.8	0.6	0.5	0.5	0.8	0.6	0.9	0.5	2.6

average fitness value after the 27^{th} and 28^{th} generation = 2.57

After the 28 the eras the hereditary procedure is stop in light of the fact that the main end condition is satisfy.

AverageFitnessafter 28^{th} gen – AverageFitnessafter 27^{th} gen = 0

Consequently the hereditary procedure is halted. The principal end condition is accomplished.

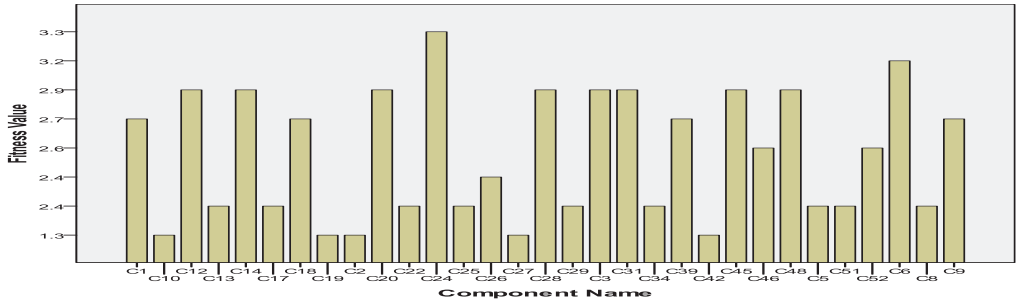

Figure 3: Fitness value of best optimal components

After assigning the attribute vectors and weight vectors to the randomly selected component from the component repository and also concluding their results by implementing the test hypothesis using chi square test so C6 and C24 find the best optimal components. The following stride is to choose the best ideal part as indicated by their rank of fitness worth.

Table 2: Best Optimal Components

Component Name	Fitness Value
C6	3.2
C24	3.3

7. Conclusion

In evaluation process possible components are chosen on the bases of keyword base search after that the selection process is started to select the best optimal component from the possible components by implement the genetic algorithms. For the effective retrieval of possible component user gives a query to select the component from the repository. In keyword based interface user type the relevant keyword for the retrieval of possible component. The interface gives all the possible results that are math with repository keywords. After matching the keywords of the interface and the component repository, the entire possible component is shown. After that genetic algorithms are implemented to get the best optimal component from the possible solution.

8. References

Bagheri, E., and F.Ensan. 2014. Reliability estimation for component-based software product lines. *Electrical and Computer Engineering, Canadian Journal of*, 37(2):94-112.

Bakshi, A. and Singh, R., 2013. Component based development in software engineering. *Int. J. Recent Tech. Eng*, *2*(1), pp.48-52.

Becker, C., and A.Rauber. 2010. Improving component selection and monitoring with controlled experimentation and automated measurements. *Information and Software Technology*, 52(6):641-655.

Chahal, K. K., and H.Singh. 2008. A metrics based approach to evaluate design of software components. In *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on* IEEE : 269-272.

Costa, N., and A. L.Coelho. 2011. Genetic and ranking-based selection of components for multilabel classifier ensembles. In *Hybrid Intelligent Systems (HIS), 2011 11th International Conference on* IEEE : 311-317.

Crnkovic, I., M.Chaudron and S.Larsson. 2006. Component-based development process and component lifecycle. In *Software Engineering Advances, International Conference on* IEEE : 44-44.

Dixit, A., and P. C.Saxena. 2009. Software component retrieval using genetic algorithms. In *Computer and Automation Engineering, 2009. ICCAE'09. International Conference on* IEEE :151-155.

Dutta, S., and S.Sengupta. 2015. Retrieval of software component version from a software version database: A graph based approach. In *Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in* IEEE : 255-259.

Fan, W., Gordon, M. and Pathak, P., 1999. Automatic Generation of Matching Function by Genetic Programming for Effective Information Retrieval. *AMCIS 1999 Proceedings*, p.17.

Fonoage, M., I.Cardei and R.Shankar. 2010. Mechanisms for requirements driven component selection and design automation. *Systems Journal, IEEE*,4(3):396-403.

Geisterfer, C. J., and S.Ghosh. 2006. Software component specification: a study in perspective of component selection and reuse. In *Commercial-off-the-Shelf (COTS)-Based Software Systems, 2006. Fifth International Conference on* IEEE : 9-15.

Hunt, J.M. and McGregor, J.D., 2010, March. Component based software engineering across the curriculum. In *2010 23rd IEEE Conference on Software Engineering Education and Training* (pp. 9-16). IEEE.

Indumathi, R., Venkateshkumar, M. and Raghavan, R., 2012, March. Integration of D-Statcom based photovoltaic cell power in low voltage power distribution grid. In *Advances in Engineering, Science and*

Management (ICAESM), 2012 International Conference on (pp. 460-465). IEEE.
Jha, P.C., Bali, S. and Kumar, U.D., 2011. A fuzzy approach for optimal selection of COTS components for modular software system under consensus recovery block scheme incorporating execution time. *Turkish J Fuzzy Syst*, *2*, pp.45-63.
Kaur, V. and Goel, S., 2011. Facets of Software Component Repository.*International Journal on Computer Science and Engineering*, *3*(6), pp.2473-2476.
Kaur, A., and K. S.Mann. 2010. Component selection for component based software engineering. *International Journal of Computer Applications*, 2(1):109-114.
Khan, A.I. and Khan, U.A., 2012. An Improved Model for Component Based Software Development. *Software Engineering*, *2*(4), pp.138-146.
Khan, M.A. and Mahmood, S., 2010. Optimal Component Selection for Component-Based Systems. In *Innovations in Computing Sciences and Software Engineering* (pp. 467-472). Springer Netherlands.
Kwong, C.K., Mu, L.F., Tang, J.F. and Luo, X.G., 2010. Optimization of software components selection for component-based software system development. *Computers & Industrial Engineering*, *58*(4), pp.618-624.
Maxville, V., J.Armarego and C. P.Lam. 2009. Applying a reusable framework for software selection. *Software, IET*, 3(5):369-380.
Pande, J., 2012. On Some Critical Issues in Component Selection in Component based Software Development. *International Journal of Computer Applications*, *46*(4).
Seker, R., and M. M.Tanik. 2004. An information-theoretical framework for modeling component-based systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 34(4):475-484.
Tomar, P., and N. S.Gill. 2013. New Algorithm for Component Selection to Develop Component-Based Software with X Model. *Lecture Notes on Software Engineering*, 1(3): 298.
Upadhyay, N., B. M.Deshpande, and V. P.Agrawal, 2011. Concurrent usability evaluation and design of software component: a digraph and matrix approach. *Software, IET*, 5(2):188-200.
Vescan, A.N.D.R.E.E.A. and Pop, H.F., 2008. Constraint Optimization-Based Component Selection Problem‖. *StudiaUniversitatis Babes-Bolyai, INFORMATICA*, *53*(2), pp.3-14.
Vitharana, P., F. M.Zahedi, and H.Jain, 2003. Knowledge-based repository scheme for storing and retrieving business components: a theoretical design and an empirical analysis. *Software Engineering, IEEE Transactions on*,29(7):649-664.