International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-3, September 2019

Recommendation-Based Component Selection for Component-Based Systems

Deepti Negi, Yashwant Singh Chauhan, Suchi Bhadula, Aditya Harbola, Amal Shankar Shukla

Abstract: Selection of reusable software components in software repositories to facilitate quality software development has always been a focal point and a big concern for software researchers. One of the most time-consuming tasks in software reusability is tracing and retrieving software components from a large repository. The selection of inapt software package can result in high cost and ultimately becomes a prime source of adverse outcome in business processes and performance of the organization. Creation of quality software depends upon the selection of the best set of components among all the alternatives present in the reusable component repository. Frequently used sets of components can be obtained by using mining algorithms. This paper proposes a component selection methodology and a method for retrieval of the optimal set of reusable components from the repository. Case-based retrieval is applied for initial filtering of components to narrow down the search space. A data mining algorithm is applied to extract the candidate set of components for a given case. Most frequent set of components aid the decision-maker to select the finest component set and also assist in suggesting the supplementary components for the case to match with the latest updates.

Keywords: component, component selection, CBSE Process, frequent mining, evaluation criteria(s).

I. INTRODUCTION

Quality and reliability are two main aspects of demand for software packages. To sustain the increasing and myriad demands of different organizations, software firms are focusing on developing customizable units that can be selected and integrated at any point of time. Tailor-made components are stored in the repository from where they can be retrieved whenever required. Selection of component from the repository is the first task that decides the path of the final software package. The selection of a wrong component may result in less reliable and low-quality software. Selection of components is a difficult task due to the availability of a large pool of components in the repository. It complicates the process of accessing and using software packages to the organizational benefits. The task of software selection is often assigned under scheduled pressure , and so the decision maker cannot stretch more time and competency to plan the detailed selection process. Therefore, appropriate selection methods are not followed properly [1].Research is required to explore and develop efficient ways for selection and evaluation of software. Some observations through the literature [2] that shows the research focus on selection and evaluation of software components are:

(1) Less work is done in the development of the decision-making framework which includes selection methodology, criteria, and techniques of software package evaluation.

(2) Less work is done on developing system/tool having an inherent knowledge of software evaluation criteria and technique. The research focus is obligatory towards increasing competence and bringing reliability and simplicity in the selection process of software packages.

(3) A number of common criteria associated with quality, reliability, cost, benefits, hardware and software requirements, stakeholders and user requirements can be focused for selection, apart from the individual functional criteria.

Most of the techniques assist decision-maker to select the best component from the repository. The components are selected based on the user given criteria(s) and requirements[3]. With the demanding market needs it is not only required to select the best component but to retrieve a complete set of components which are most recent in the market for any specific case. Thus it becomes a necessity to find the set of components which aid the decision-maker to select the premium component plus some additional components that can be added to user requirement set to update the product to its latest. Frequent Pattern mining techniques[4][5][6] can be applied to find the set of frequently occurring components. The organization of this paper is as follows: A basic introduction of the software component selection process is given in Section 2. Section 3 gives a brief summary of the related work done in the area of software component selection. Section 4 describes the proposed methodology and component set selection approach. Finally, Section 5 is conclusion and the future work.

Manuscript published on 30 September 2019

* Correspondence Author

Deepti Negi*, School of Computing, Graphic Era Hill University, Dehradun, India. Email:deeptine@gmsil.com

Yashwant Singh Chauhan, Computer Science and Applications, GBPEC, Ghurdouri, Pauri, India. Email: yashwantschauhan@gmail.com

Suchi Bhadula, Computer Science Engineering, Graphic Era University, Dehradun, India. Email: bhadula.suchi@gmail.com

Aditya Harbola*, School of Computing, Graphic Era Hill University, Dehradun, India. Email:adityaharbola@gmail.com

Amal Shankar Shukla, Department of Computer Applications, Graphic Era University, Dehradun, India. Email: amalshankarshukla@gmail.com

II. COMPONENT SELECTION PROCESS

Software component selection process is mainly divided into two phases, first is to incorporate the criteria(s) for election and second is the calculation of the final score of each alternative based upon the selection criteria. This final score ranks the alternatives and helps to identify the best suitable component. The component selection process is a multi-criteria decision-making process and requires an appropriate decision-making algorithm for score calculation. Criteria based selection process such as 'six sigma' is a formal process of component selection[7]. However, informal approaches are also available for the process of component selection. [8]. Number of times informal methods based on experience, Hands-on-Trial and Customer Recommendations for COTS component selection are preferred.

Assessment and selection of components is a difficult decision-making process which is very time-consuming. Selection of inapt package can result in high cost and ultimately becomes a prime source of adverse outcome in business processes and performance of the organization. Selection of best possible component from the repository is always a major concern for decision-makers.

A. Component Selection Problem

Component Selection Problem assists the designer and decision-maker to make sensible choices of component sets that result in quality software. The selection problem can be further categorized as Simple Component Selection Problem(SCSP) and Criteria-Based Component Selection Problem(CCSP). In the first category, components are selected as per the requirements are given by the user. In second category selection process depends upon selected criteria(s).

In SCSP components are selected from the repository in such a manner that their composition satisfies the requirement set given by the user. SCSP can be explained as, RS = {r1, r2, ..., rn},is a requirement set and CS= {c1, c2, ..., cm} is the set of components in the repository. A subset of requirements from the set RS, satisfied by a single component ci, is represented as RSci = {ri1, ri2,..., rik}. The selection process aims at selecting a set of components such that each requirement rj from the set RS can be assigned a component ci. CCSP deals with selecting a set of components such that their composition achieves a provided set of requirements by following criteria. The aim is to select the best possible set of components with minimum cost.

III. RELATED WORK

Software component selection is one of the most significant processes towards developing quality software. Different authors provide various approaches and methods to select the best possible set of components from the repository. A brief review of the methods and techniques is illustrated in the following section.[9][10][11][12][13][14][15][25]

A. Weighted Scoring Method(WSM)

One of the oldest, common and manual method for evaluating and selecting components is Weighted Scoring Method. This method considers multiple criteria for selection of components and thus follows Multi-Criteria Decision Making (MCDM) approach. WSM technique can be used if there are 'c' numbers of criteria and 'n' numbers of components in the repository. Steps in WSN are given in figure 1.

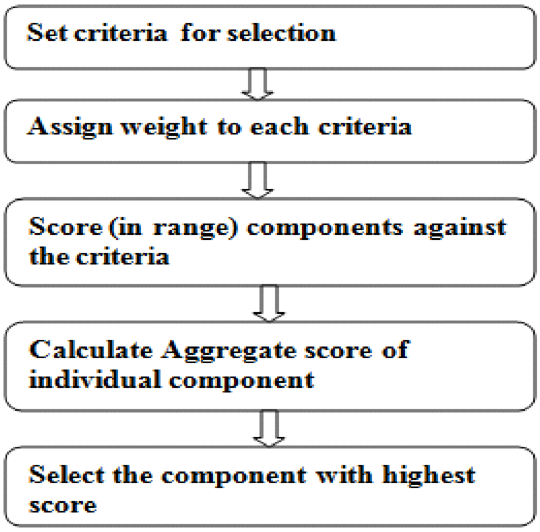


Fig. 1. Steps in Weighted Scoring Method

WSM technique is explained through an example in table 1. Suppose the user grants the following four selection criteria for software packages: user-friendly, least response time, flexibility and low cost. Next, he assigns a weight to the given four criteria according to their relative importance on a scale of 1-4. With the help of the given formula individual score of each component is calculated.

$$S(Ai)=\Sigma WjSij$$

here j=1,2,…, n; Wj is the weight of jth criterion; Sij measures the score (performance) of a candidate component (Ai) on selected criteria.

The table shows that component C5 is the best one to be selected among all choices as it scores the highest value.

.

Table -I: Component Selection using WSM

Selection Criteria	**Weight**	**Components**					
		C1	*C2*	*C3*	*C4*	*C5*	*C6*
User friendly	30	2	3	2	4	5	2
Least response time	20	4	5	4	3	2	3
Flexibility	25	1	3	4	2	4	4
Low cost	25	3	5	4	2	5	1
Total score		**2.4**	**3.9**	**3.4**	**2.8**	**4.15**	**2.45**

Collier. K et al [16] applied WSM technique in an efficient manner for selection of the best data mining tool among three alternatives available. WSM is simple to understand and easy to use method. The problem comes when there is a large pool of alternatives, as WSM is a manual process. Also if the requirement changes in between, the score and evaluation criteria of each component changes and thus require upgrading before the final calculation. When a large number of criteria are defined, it becomes very difficult to assign weight and a common numerical scaling is required to obtain the final score.

B. Analytical Hierarchical Process (AHP)

Another method, given by Dr. Thomas Saaty named AHP is used for multi-criteria decision making. It has been widely used in areas of education, banking, manufacturing, etc. Cangussu J. W et al [17] utilize the AHP process effectively in analysis and ranking of data compression technique. Both quantitative, as well as qualitative factors, are considered for selection of best alternative. This method assists decision-makers to describe the decision problems into hierarchical order. The hierarchical order of decision problem enables the user to understand the flow of solution but it takes lots of time. A large number of mathematical calculations and pair-wise comparisons are performed which results in a time-consuming process. If any change occurs in requirements, this complete exercise has to be repeated. AHP process is based upon three basic principles of decomposition, comparative judgments, and synthesis of priorities. The steps are given in figure 2.

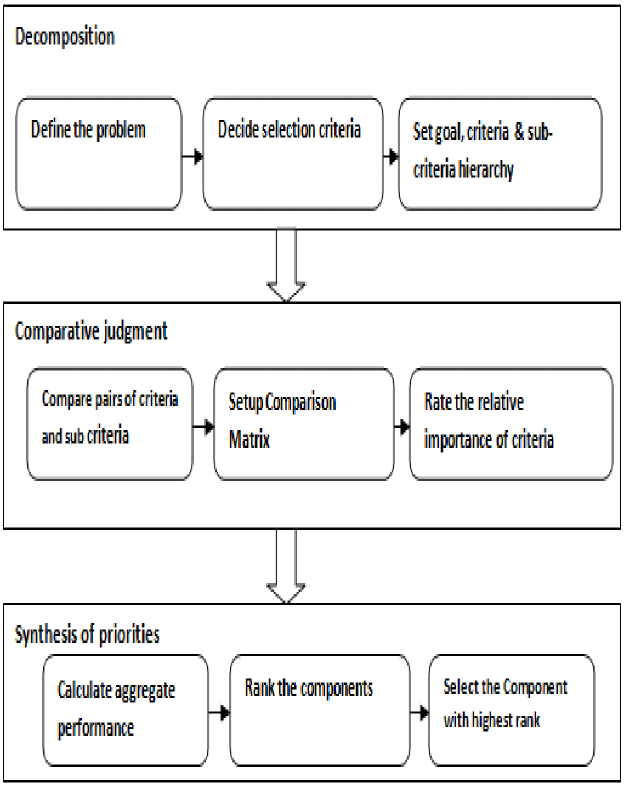


Fig. 2. Steps in AHP process

AHP also suffers from the rank-reversal problem. However, a multiplicative-formula for aggregation is used for rectification of this problem [18].

C. Knowledge - Based System

A blend of rule-based reasoning(RBR) and case-based reasoning(CBR) technique is the key concept behind the idea of the knowledge-based system proposed by Jadhav. A et al [19]. CBR and RBR are two fundamental reasoning processes of a knowledge base system(KBS). This selection technique helps in arranging the user given knowledge in a systematic manner and provide a tool for the evaluation and selection of software packages to decision-makers. The methodology is elucidated in Figure 3.

RBR assisted the user to analyze and formulate the evaluation criteria in simple logical statements. Software packages are organized in terms of feature and feature value based upon the user requirements.CBR compares criteria based requirements against the desired software components to the candidate software packages. These candidates were stocked as 'cases' in the case-base of the repository system. A result set is formed consisting of the candidate components which are ranked based on their similarity score.

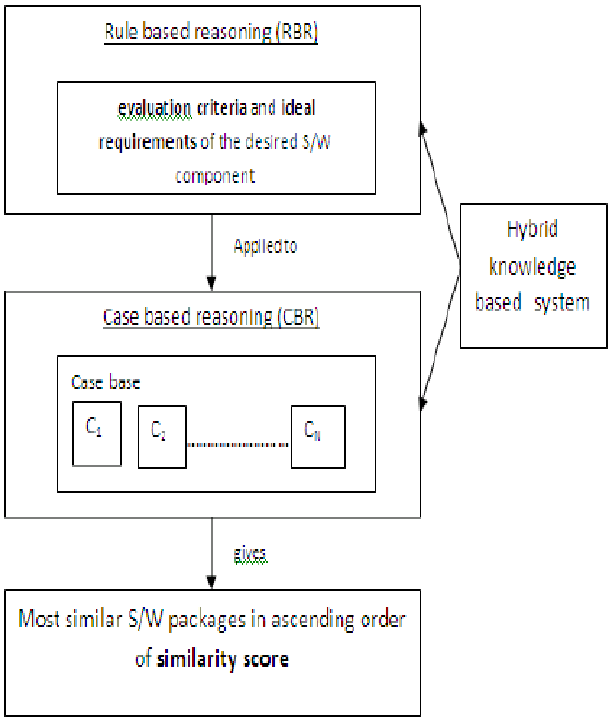


Fig. 3. HKBS System

A comparative study carried out [20] between the three approaches, WSM, AHP and HKBS shows that HKBS method is comparatively better than AHP and WSM.

D. Automated and Semi-Automated Approaches

Maxville. V et al [21] proposed and apply classification approach in the component selection process. Based on the project calculations, components are accepted and assigned to a class. Context-driven component evaluation (CDCE) approaches are also used for selection and automation of the repeatable process. The CDCE process minimizes the time and effort in the component selection process by using Artificial Intelligence techniques. AI techniques consider the conflicts among the dependencies of the component's attributes. Special classifiers are designed and trained with AI techniques to identify and select appropriate components that fit the user requirements. This methodology was an effective alternative to AHP and WSM as it can be applied to a large number of components. An extra overhead with this approach is the extra time consumed in providing the precise specification of the preferred component. The semi-automated technique proposed [21] can only be used to narrow down the choices of alternatives but is not fit for selection of the best solution.

Retrieval Number: C5622098319/2019©BEIESP
DOI:10.35940/ijrte.C5622.098319
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Another model was given by Abraham. B.Z et al [22] based on the concept of Swarm Intelligence (S.I). Some initial ideal requirement specification written in XML is documented. The components are selected in accordance with the given specification. Categorization of the component specification is done as static and dynamic. Dynamic characteristics are changeable whereas static characteristics remain the same throughout. Pheromone tracking is used to identify and select the best component. Positive and negative feedback is taken from the user and accordingly the pheromone value is updated. The value is increased for positive feedback and decreased negative feedback. The components with the highest pheromone value are considered as the best for selection. The benefit of this approach is its diversity in fields of component selection, services, resources, etc .

E. Fuzzy Based Approach

Fuzzy based selection technique or approach enables decision-makers to advance decision making procedure by handling the imprecision and ambiguity in human decision making. Still, computation of appropriateness of index value and ranking for all the components is a complex task.

IV. PROPOSED WORK

The proposed work is alienated into two divisions, first is a methodology that presents a well-defined road map towards software-component selection and is flexible enough to promote a personalized selection. Second is an approach named Recommendation-Based Component Selection Approach, aiming towards selecting the best in-use set of components from the repository. The proposed methodology is a pathway towards the development of quality software by selecting the most frequent sets available in the repository. This approach helps in predicting the supplementary components which can be integrated with the user-specified criteria/requirements to make the software attuned with the latest update of their similar counterpart.

A. Methodology for Component Selection

A series of steps are followed by decision-makers to select an appropriate set of components from the reusable repository. A methodology depicts the sequence of steps that can be pursued in order to attain a solution. Methodology generalized the way towards reaching the solution, but should not be rigid[23]. Depending upon the preference of decision-maker and organizational requirements, a methodology can be customized. In this paper, we have proposed a methodology to facilitate the decision-maker a step by step process of selecting components. Fig. 4 depicts the general steps followed during component selection methodology which are explained below.

Collect Requirement Set: First step is to assemble a complete set of user requirements including all the functional and nonfunctional requirements. Absolute set of requirements is the foundation for the final product.

Analyze/Prioritize/Set Requirements into Criteria: This phase is the contemplation of myriad ideas among the decision-makers to transform the requirements into criteria and set their priority.

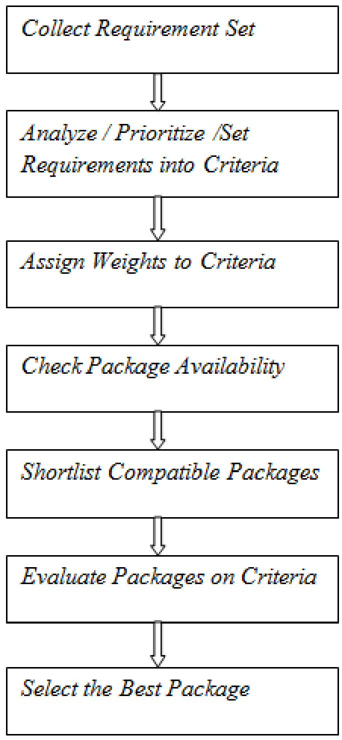
Assign weights to criteria: In this phase weights are assigned to the selected criteria on a scale of a given range (say 1 to 100) in order of priority.

Check Package Availability: This phase checks for all the components available in the repository that matches the specified criteria set.

Shortlist Compatible Packages: This is a phase that selects the minimal set of components among the sets of candidate sets selected in the previous phase. After shortlisting only matched and closely matched cases are the candidate set for final selection.

Evaluate Packages on criteria: The selected packages are evaluated by applying some formula or approach on the weights assigned to criteria. After evaluation, the packages are ranked in the order of their final scores.

Select the Best package: Among the ranked components, the one with the highest score is considered best and is selected.



.Fig. 4. Methodology for component selection

B. Recommendation-Based Component Selection Approach

AHP and WSM techniques are not considered as driven by the requirements and hence they are not sufficient for software selection decision making [24]. Our proposed approach will help the decision-makers to select the best possible set of components that matches the user criteria, and set of components which can be suggested to the client for the advancement of the overall functionality of the product. Component set is a collection of components which are frequently used as a set in most of the cases that match with the user given case. Frequent set of components helps the decision maker to enhance their build scope by selecting the best possible combination of the software components which are most recent in use. This key idea of the approach is to utilize the collective efforts of rule-based and case-based selection. Our Proposed Approach is explained in the following section.

Requirement set collected from the client is transformed into a fixed set of criteria by a team of knowledge workers after profound analysis. Prioritization of criteria and assignment of weights to each criterion is the next step in the process. The role of a rule based expert system is to frame the rules-based upon the criteria and their respective priority. The case-based expert system is a repository that contains a pool of cases, stored as rules, and a list of components utilized in the case. This expert system stores cases based upon the past experiences and knowledge of decision-makers. The two expert systems, Knowledge-Based Expert System and Case-Based Expert System, are matched against each other to generate a case-profile. The case-profile is a data structure created as a result of the similarity between the rules stored in KBES and CBES with the fields case name, case id and set of components. Figure 5 explains the proposed approach.

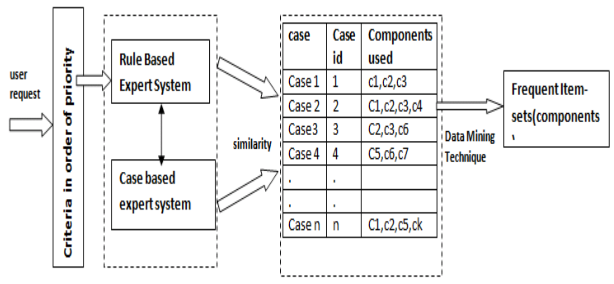


Fig. 5. Requirement-Driven Knowledge-Based Component Selection Approach

The data structure (case-profile) generated stores three fields: first is the case name, which is the description of the pre-existing case, second is case-id which is a unique serial number given to the case and third is components, a set of components utilized in the case. Table 2 shows a sample case-profile.

Case name	Case id	Components
AAA	1	c1,c3,c4
BBB	2	c1,c5,c7,c8
CCC	3	c1,c3,c4
DDD	4	c1,c3,c4,c5

Table-II: Data Structure for Case-Profile

The case profile is an input for Apriori algorithm with case id as transaction id and component sets as item sets. Apriori property says, if an item set is frequent, all its subset will be frequent. If an item-set is below the minimum support threshold, minsup, then it is not frequent. If I is a nonfrequent item set and A is added to I, then I U A will not be a frequent set either, that is, P(I UA) < minsup. This is called a downward closure property. Apriori algorithm works in two phases considering T as a set of transactions in the database

Phase I (SCAN):. In this step, we first generate the candidate item set. This is done by pairing the item sets. Then we count the number of times these item sets occur in each transaction.

Phase II (PRUNE): Since the number of candidate item sets can be large in number, this leads to heavy computation. To reduce the size of the candidate item set, we prune those item sets which occur below the minimum support.

V. STEPS FOR FREQUENT SET GENERATION USING APRIORI ALGORITHM

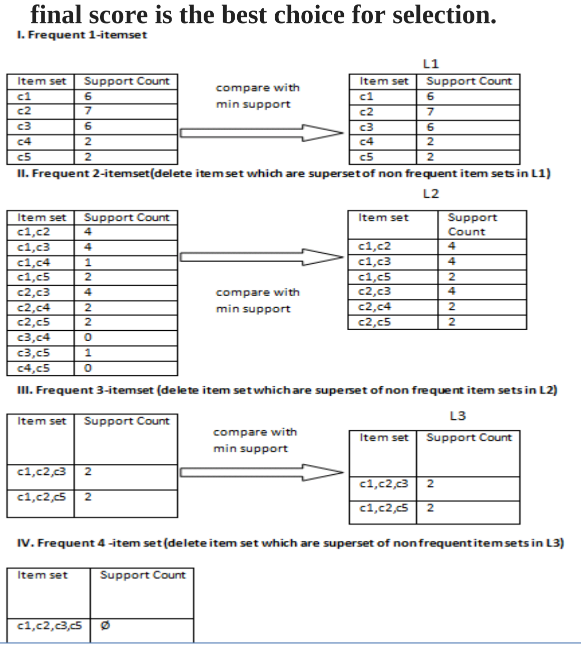
Let Table 3 shows an input case profile for Apriori Algorithm. Considering minimum support value (threshold) 2, we can locate the frequent set which is above this minimum support. The frequent item set generation process is depicted in figure 6.

TID	Component Set
T1	c1,c2,c5
T2	c2,c4
T3	c2,c3
T4	c1,c2,c4
T5	c1,c3
T6	c2,c3
T7	c1,c3
T8	c1,c2,c3,c5
T9	c1,c2,c3
T1	c1,c2,c5

Table-III: Input case profile

As, Support count for frequent - 4 item set = Ø, the algorithm terminates. From the result generated above, we found that { c1,c2,c3,c5} is the most frequently used set of components that matches the user given criteria based requirement set. Hence it is the best choice for selection. It may happen that the process results in more number of frequent sets which are above the selected threshold value. Such cases can be handled by applying the formula given below

Step 1: The total score for alternative Ci is calculated as:
S(Ci) = ∑WjSij , where sum is over j=1, 2, . . ., n;
Where , Wj is the relative weight of jth criterion;
Sij is the score measured on jth criteria by component Ci.
Step 2: The final score for a set of k components can be calculated as
F(Si)= ∑ S(Ci) , where sum is over 1, 2, . . ., k;
Among all the alternatives, the set with highest value of final score is the best choice for selection.



Item set	Support Count
c1	6
c2	7
c3	6
c4	2
c5	2

L1

Item set	Support Count
c1	6
c2	7
c3	6
c4	2
c5	2

Item set	Support Count
c1,c2	4
c1,c3	4
c1,c4	1
c1,c5	2
c2,c3	4
c2,c4	2
c2,c5	2
c3,c4	0
c3,c5	1
c4,c5	0

L2

Item set	Support Count
c1,c2	4
c1,c3	4
c1,c5	2
c2,c3	4
c2,c4	2
c2,c5	2

Item set	Support Count
c1,c2,c3	2
c1,c2,c5	2

L3

Item set	Support Count
c1,c2,c3	2
c1,c2,c5	2

Item set	Support Count
c1,c2,c3,c5	Ø

Fig. 6. Apriori Algorithm for generation of frequent item set

Let { c1,c2,c3,c8},{c1,c2,c4,c8} and {c2,c3,c5,c8} are three frequent sets that are above the threshold value. Among the three sets, set 3 is having maximum value of final score which is 10.4, so is the best option for selection. The process is illustrated in table 4.

VI. RECOMMENDATION OF SUPPLEMENTARY COMPONENT

From the above set of the frequent item set, it is observed that component c8 is a part of each of the resultant set. Though it is not satisfying any of the user-specified criteria but is a participant in most of the similar cases. As this component is frequent in most of the similar cases, the decision-maker can suggest the client to consider and add component c8 to enhance the selection criteria.

Addition of this component will help the client to be updated with the latest and similar complements of his software.

Table-IV. Component Selection using Recommendation-Based Selection

Selection Criteria	Weight	Components											
		SET 1				SET 2				SET 3			
		C1	C2	C3	C8	C1	C2	C4	C8	C2	C3	C5	C8
User friendly	30	2	3	2	0	2	3	2	0	3	2	4	0
Least response time	20	4	5	4	0	4	5	3	0	5	4	2	0
Flexibility	25	1	3	4	0	1	3	4	0	3	4	1	0
Low cost	25	3	5	4	0	3	5	1	0	5	4	5	0
Total score		2.4	3.9	3.4	0	2.4	3.9	2.45	0	3.9	3.4	3.1	0
Final Score		**9.7**				**8.75**				**10.4**			

VII. CONCLUSION

Component-Based software Engineering, in contrast with building from scratch approach, allows selecting and integrate already existing reusable components to reduce both complexities of the software and cost, and improves quality. One of the most time consuming task in CBSE process is the evaluation and selection of the software packages. Selection of an appropriate methodology is the first step towards the development of quality software. It aids decision-maker to follow steps for selection of the best set of components. Our paper proposes a methodology of the selection process of components from an existing repository. We have also proposed an approach to find the frequently used set of components based upon similarity in case-based and rule-based expert systems. This approach assists decision-maker to select the best set of available components based upon user given criteria(s) in the order of their priority. Most frequent set of components aid the decision-maker to select the finest component set and also assist in suggesting the supplementary components for the case to match with the latest updates. In future, we can also apply the advanced algorithm to improve efficiency.

REFERENCES

1. Kontio, J., Caldiera, G., Basili, V.R., 1996. Defining factors goals and criteria for reusable component evaluation. In: CASCON'96 Conference , Toronto, Canada, November 12–14.
2. Jadhav, A.S., Sonar, R.M., 2009. Evaluating and selecting software packages: a review. Information and Software Technology 51, 555–563.
3. Arditi, D., Singh, S., 1991. Selection criteria for commercially available software in constructing accounting. Project Management 9 (1), 39–44.
4. Fayyad U M, Piatetsky-Shapiro G, Smyth p, The KDD process for extracting useful knowledge from volumes of data[J], Communication of the ACM, 1996. 39 (11)
5. J. Han, M. Kamber and J. Pei. Data Mining Concepts and Techniques. 3/e Morgan Kaufmann Publishers, ISBN 978-0-12-381479-1, 2012.
6. R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large database. In: Proc ACM SIGMOD Intl. Conf. Management Data, 1993.
7. A. Cechich and M. Piattini, "Challenges Setting a Process to Manage COTS Component Selection", Proceedings of the First ICSE Workshop on Models and Processes for the Evaluation of COTS Components, 2004, IEE Press, 49--51.
8. J. Li, F. Bjornson, R. Conradi and V. Kampenes, "An empirical study on COTS component selection process in Norwegian IT companies", Proceedings of the First ICSE International Workshop on Models and Processes for the Evaluation of COTS Components, IEE, Edinburgh, Scotland (2004), pp. 27–30
9. Nakkrasae S., Sophatsathit P. and Edwards W. R. "FUzzy Subtractive Clustering based Indexing Approach for Software Components Classification", International Journal of Computer and Information Sciences, vol.5, no. 1, pp. 63-72,2004.
10. Stylianou C. and Andreou A. S., " Hybrid Software Component Clustering and Retrieval Scheme using an Entropy-based Fuzzy k-modes Algorithm"proc. 19th IEEE International Conference on Tools with Artificial Intelligence, 2007, pp.202-209.
11. Ossadnik, W., Lange, O., 1999. AHP-based evaluation of AHP-software. European Journal of Operational Research 118, 578–588.
12. Patel, N., Hlupic, V., 2002. A methodology for the selection of knowledge management (KM) tools. In: 24th International Conference on IT Interfaces , Cavtat, Croatia.
13. Perini, A., Ricca, F., Susi, A., 2009. Tool-supported requirements prioritization: comparing the AHP and CBRank methods. Information and Software Technology 51, 1021–1032.
14. Pal, K., Campbell, J., 1997. An application of rule based and case based reasoning within a single legal knowledge-based system. The Database for Advances in Information Systems 28 (4), 48–63.
15. Godse, M., Sonar, R., Jadhav, A., 2009. A hybrid approach for knowledge-based product recommendation. In: Third International Conference on Information Systems Technology and Management March 12–13, 2009 , Communications in Computer and Information Science, Springer, vol. 31, pp. 268–279.
16. K.. Collier, B. Carey, D. Sautter and C. Marjaniemi, "A Methodology for Evaluating and Selecting Data Mining Software",Proceedings of the Thirty Second Hawaii International Conference on System Sciences, 1999
17. J. W. Cangussu, K. C. Cooper, and Eric W. Wong, "Multi Criteria Selection of Components Using the Analytic Hierarchy Process", Proceedings of the 9th International Symposium (CBSE), pp. 67- 81, Springer, 2006.
18. E. Triantaphyllou, Multi-Criteria Decision Making Methods: A Comparative Study, Kluwer Academic Publishers, 2001.
19. A. S. Jadhav and R. M. Sonar, "A hybrid system for selection of the software packages", Proceeding of first international conference on emerging trends in engineering and technology ICETET,2008, IEEE Xplore, 337-342.

20. A. S. Jadhav and R. M. Sonar, "Analytic Hierarchy Process (AHP), Weighted Scoring Method (WSM), and Hybrid Knowledge Based System (HKBS) for Software Selection: A Comparative Study", Second International Conference on Emerging Trends in Engineering and Technology, ICETET, 2009, IEEE.
21. V. Maxville, J. Armarego, and C.P. Lam, "Intelligent Component Selection", proceedings of 28th annual international computer society and application conference, COMPSAC, 2004.
22. B. Z. Abraham , J. C. Aguilar, "Software Component Selection Algorithm Using Intelligent Agents", Proceedings of the 1st KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications, May 31-June 01, 2007, Wroclaw, Poland.
23. Blanc, L.A., Korn, W.M., 1992. A structured approach to the evaluation and selection of CASE tools. In: ACM.
24. Ncube, C., Dean, J.C., 2002. The limitations of current decision making techniques in the procurement of COTS software components. In: Proceedings of the First International Conference on COTS-based Software System , Orlando, February 2002, pp. 176–187.
25. D. Negi, Y. S. Chauhan, P. Dimri and A. Harbola, "An Analytical Study of Component-Based Life Cycle Models: A Survey," 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, 2015, pp. 746-750. doi: 10.1109/CICN.2015.152

AUTHORS PROFILE

Deepti Negi MCA, PhD(P) Software engineering, Component based software engineering 1. An Analytical Study of Component-Based Life Cycle Models: A Survey , IEEE 12-14 Dec. 2015 DOI: 10.1109/CICN.2015.152
2. Green computing research challenges: a review , IJARCSSE, January 2013 ,
Membership: CSI, IEEE

Yashwant Singh Chauhan MCA, DPhil, PhD Fractals, Computer Graphics, Software engineering, Membership: CSI, IEEE
1. Yashwant S Chauhan, Rajeshri Rana and Ashish Negi. "New Multicorns of Sine Function", International Journal of Mathematics and Computational Methods in Science & Technology, Vol. 1, No.4, 73-81, 2011. ISSN 2249 8915.
2. Green computing research challenges: a review , IJARCSSE, January 2013
3. Yashwant S Chauhan, Rajeshri Rana and Ashish Negi. "A Computational Study of Logarithmic Function For Mann and Ishikawa Iterates", World Journal of Science and Technology 2011, 1(10): 92-101.ISSN: 2231 – 2587.

Suchi Bhadula MCA, MTECH (CSE) PhD(P) Software engineering, Component based software engineering
Membership: CSI, IEEE.

Aditya Harbola MCA, MTECH (CSE) PhD(P) Software engineering, Machine learning
1. An Analytical Study of Component-Based Life Cycle Models: A Survey , IEEE 12-14 Dec. 2015 DOI: 10.1109/CICN.2015.152
2. Green computing research challenges: a review , IJARCSSE, January 2013
3. Improved intrusion detection in DDoS applying feature selection using rank & score of attributes in KDD-99 data set, IEEE 14-16 Nov. 2014 DOI: 10.1109/CICN.2014.179

Dr. Amal Shankar Shukla, MCA, PhD
Membership: CSI, IEEE Software engineering, E Governanve, ERP