ISSN:0975-9646

Loveleen Kaur et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3739-3742

Software Component Selection techniques - A review

Loveleen Kaur & Dr. Hardeep Singh

Dept. of Computer Science Engineering
Guru Nanak Dev University
Amritsar, Punjab, India.

***Abstract*— Component Based Software Engineering(CBSE) relies on the premise of reuse and aims to develop systems by selecting appropriate off the shelf software components and integrating them in order to achieve its desired task. Usage of COTS component ensures faster time-to-market. At the same time, component-based software introduces risks like unknown quality properties of the components in use, that can inject harmful side effects into the final system. Therefore, component selection is one of the most tedious and challenging tasks of CBSE and involves simultaneous consideration of multiple selection criteria as per the user requirements.**

This is a review paper which aims to study some of the various formal software selection techniques that have been introduced in the literature so far, pointing out the benefits and limitations of each and what could be the future work possible with respect to the already introduced techniques.

***Keywords*— software component selection, component based software engineering, multi criteria decision making, formal software selection process, software component.**

I. Introduction

Component based software engineering (CBSE) is the reuse-based, sub-discipline of software engineering which defines, implements and integrates independent software components into a system. In the words of Ian Sommerville [1] "CBSE emerged from the failure of object-oriented development to support effective reuse. Single object classes are too detailed and specific. Components are more abstract than object classes and can be considered to be stand-alone service providers".

Component based software development stresses reuse of already developed commercially off the shelf available components i.e. COTS base development. It emphasizes on 'buy, don't build philosophy'. Reusing previously developed components in developing software has many benefits like reduced cost and lesser time to market . Further, a repeated use of a component will result in its repeated testing in various domains. This will definitely result in an increase in the quality of the component as well as the system where it is being used. The term 'Component' here could refer to a software package, a web service or a module that has a set of related functions and is capable to performing a task independently and in collaboration with other components.

CBSE is used in the following two cases:

1) One has to build a system by integrating various COTS components.
2) One wants to integrate a COTS component to an existing system not necessarily built using a CBSE approach.

In both the cases, careful selection of desired component from a COTS repository need to be done because it plays a vital role in the success of the final system. The selection of components is often considered as a tedious task in component based software engineering, as there are complex technical, legal, and business considerations that need to be taken care of as development proceeds. Since the selection of a suitable component out of a pool of candidate components is not based on a single evaluation criteria, difficulties might arise in selecting the appropriate component because the selection criterias might conflict with each other. The decision making approaches proposed to address this problem are called multi-criteria decision making (MCDM) methods.

This paper is organized as follows: Section 2 provides with some basic introduction to the software component selection process. Section 3 aims to study some of the vital manual and automated techniques of software component selection that have been introduced in literature so far, examining the benefits and limitations of each. Finally, Section 4 consists of conclusion and the future work.

II. Software Component selection process

Software component selection process has been primarily composed of the incorporation of a selection criteria(s) and the calculation of the score of each alternative with respect to the selection criteria(s) which determines which alternative suits our needs the most. Since software component selection process is a multi-criteria decision making process, the score of each alternative is calculated using an appropriate decision making algorithm. This is a formal process of software component selection eg:- The 'six sigma' approach of software component selection[2].

However, there is an informal approach to the software component selection process as well [3]. At times, software engineers prefer informal methods of COTS component selection like Experience-Based component selection, Hands-on-Trial based component selection and the Customer Recommendation based COTS component selection. Out of the various informal approaches, the hands-on-trial methodology is considered as an important and necessary method of COTS component selection since it leads to trust in and careful examination of the COTS component by testing the COTS component locally. This paper focuses on the various formal techniques of COTS component selection.

III. RELATED WORK

Formal software component selection, in literature, has been primarily divided into manual and automated approaches.

A. Manual approaches to component selection

1) Weighted Scoring Method(WSM): Weighted scoring method is one of the oldest techniques used for evaluation and selection of the software packages. Since it is a MCDM approach, this technique is applied to those situations in which there are 'n' number of candidate components and 'm' number of evaluation criterias. For selecting the best component out of the various candidate components, using WSM, five steps are taken as shown in Fig 1.

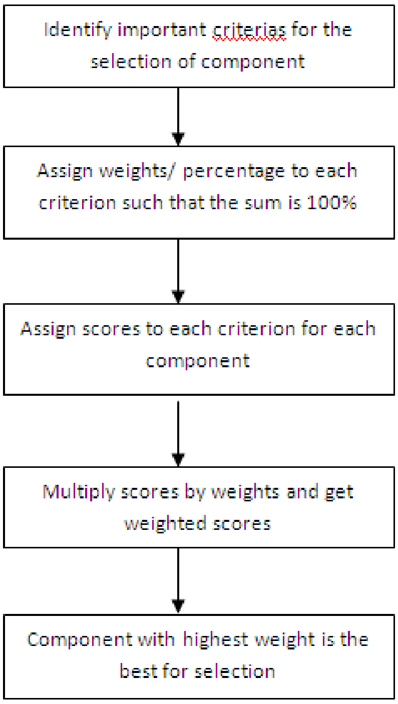

Fig 1: The WSM process

Consider a situation in which we have to select a component on the basis of 5 selection criteria:- greatest ease of use, having least system overhead, least response time, least cost and high customizability. Weights are assigned accordingly so as to state the relative importance of each selection criteria. As it can be seen, least cost is the most important selection criteria. The four candidate components are assigned their performance on each criteria on the scale of 1-5. The final score of each component is calculated using the following formula

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

Where sum is over j=1,2,…, n; Wj is weight of jth criterion; Sij is score that measures how well a candidate component (A_i) performs on a selection criterion. Thus in the example given below (Table 1) we can see that Component 1 (C1) is the best for selection according to the selection criterias since it has the highest score out of the four alternatives.

Selection criterias	Weights	Components			
		C1	*C2*	*C3*	*C4*
Ease of use	20	5	4	3	5
System overhead	10	4	2	2	2
Response time	5	1	5	5	3
Cost	30	5	5	4	1
Customizability	35	5	4	3	4
Final score of each component		**4.7**	**4.15**	**3.3**	**3.05**

Table 1: Selection of best component using WSM

Note :- in the above example since a less cost, lesser response time and less system overhead is desired, the score '5' indicates that the alternative has least cost and least system overhead. Similarly score of 1 indicates highest cost and highest system overhead.

Many techniques in the literature have used the WSM method for selection of component and software packages. However, Collier. K et al [4] most effectively used the WSM technique in their work for selecting the best data mining tool out of three alternatives. Though this selection technique is extremely simple and convenient to use, the main disadvantage is that it cannot be used for a large number of alternatives since it is a manual process. Also, if the user requirements change at the last moment, the score of each alternative with respect to each evaluation criteria changes and this change needs to be upgraded before calculating final score.

2) Analytical Heirarchical Process(AHP): AHP is the other multi-criteria decision making method apart from WSM and was developed by Dr. Thomas Saaty during the 1970s. It has been applied in wide variety of applications in various fields like banking, education and manufacturing. This method allows consideration of both qualitative and quantitative factors in selecting the best alternative. In short, it uses an 'eigen value' approach to derive ratio scales from paired comparisons. In general, paired comparison takes this form: "How important is criteria C_i relative to criteria C_j?.The performances of the various alternatives are caliberated with the help of a numeric scale which ranges from 1/9 to 9. A score of 1/9 indicates 'least valued than', a score of 1 indicates 'equal to' and a score of 9 indicates 'most important than'. The AHP methodology is based on three principles: decomposition, comparative judgments; and synthesis of priorities.

(a) The decomposition step involves defining the problem, deciding the criteria that influences the selection process and structuring the problem in a hierarchy of different levels constituting goal, criteria, sub-criteria and alternatives.

(b) With the comparative judgments, users are required to set up a comparison matrix at each level of hierarchy by comparing pairs of criteria or sub-criteria. Next step is to rate the relative importance of these criteria using pair-wise comparisons. This includes the comparison of each element in the corresponding level and its appropriate value on the numerical scale.

(c) The final stage is to calculate aggregate performance value for each alternative and ranking of alternatives according to their performance.

AHP has been widely used to effectively in various cases where multi criteria decision making is required. However, it has been most effectively used by Cangussu J. W et al [5] to analyze and rank the most effective data compression technique. The criteria(s) for selection in the above mentioned paper have been restricted to non-functional ones although the future work suggests that this technique can be extended to include multiple functional criteria(s) for the selection process as well. Though this technique is more organized in terms of its structure its limitations lie in the number of pair-wise comparisons (and therefore time) required. In addition to this, AHP also suffers from rank-reversal problem which can be rectified using the multiplicative-formula for aggregation[6].

B. Automated/ Semi-Automated approaches to component selection

Maxville. V et al [7] approached the component selection process as a classification problem in which every component could be assigned a class based on its calculated acceptability to the project. The concept of context driven component evaluation (CDCE) approach, for the purpose of automating a repeatable process of component assessment (i.e component selection and component evaluation), was introduced in this paper. The CDCE process took the various conflicting dependencies of the component's attributes into consideration and used the Artificial Intelligence (A.I) as a medium to reduce the time and effort involved in the component selection process. Since this was a classification approach, classifiers were trained to recognize suitable components based on appropriate AI techniques. Two AI techniques were selected to generate the classifiers – C4.5 and Artificial Neural Networks(ANN). These classifiers were evaluated on the dataset with the help if a machine learning tool – Weka. The data needed to train the two classifiers was generated from an ideal specification of the required component. This ideal specification was written in XML. Both the C4.5 algorithm and ANN were able to correctly classify more than 99% of the training data set. However, the C4.5 classifier gave better results as compared to ANN when an unseen dataset was provided to it. The interplay of attributes was also taken into account to test the accuracy of the classifiers. Both the classifiers gave 100% accurate classification of the data in this situation. This proposed methodology was an effective alternative to the usual aggregation based component selection techniques (AHP and WSM) and could to be applied to large numbers of components. Also the C4.5 and neural network classifiers recognised suitable components with a high level of accuracy. The only disadvantage of this approach was the little overhead involved in providing the accurate specification of the desired component for any selection task.

Abraham. B.Z et al [8] developed a model exhibiting an intelligent selection of software components using the concept of Swarm Intelligence (S.I). The selection process selects the most suitable component among a group of components according to some initial ideal requirements or specifications written in XML. The ideal components specification and the candidate component's specification is segregated into static and dynamic categories. Dynamic characteristics are those that could change over time eg:- performance of a component when applied to a specific platform. Static characteristics are those that do not change over time eg:- component's name. Since this technique is based on S.I, it relies on the pheromone tracks to recognize best component, according to our needs, adding or evaporating pheromone each and every time the component's performance is evaluated in a particular domain. This pheromone value will be stored in the component's profile. The authors proposed an algorithm 'A' used to select the best alternative taking into account both negative and positive feedback from the component's performance. Positive feedback will result in an increase in the pheromone value of the component and a negative feedback will result in some pheromone evaporation of the component exhibiting that the component does not show adequate performance in a particular environment. Lastly, component with the highest pheromone value is selected. The main advantage of this approach was that it is general enough for been replicated with different types of requirements and so this model can be used not only for component selection but also for services, resources, etc.

Jadhav. A et al [9] presented a technique incorporating the combined efforts of rule based reasoning (RBR) and case based reasoning(CBR) for the effective selection of software components. CBR and RBR are two radical reasoning methods of a knowledge base system(KBS). KBS provides a mechanism to systematically arrange the knowledge and produce a tool that assists decision makers in evaluation and selection of the software packages. The methodology proposed by the authors is explained in Fig 2.

- RBR assisted the user in deciding the evaluation criteria and capture the requirements of the ideal software package. The rules in the presented system were written in a simple IF-THEN-ELSE format. User requirements of the software package are collected in the form of feature and feature value.
- CBR compared the ideal user requirements of the desired software to the candidate software packages. These candidates were stored as 'cases' in the case-base of the system.
- Result set of the system presented the candidate components ranked on the basis of the similarity

score. This similarity score stated so as to how well each candidate software meets the needs of the desired software package.

A comparison [10] was also conducted between the WSM, AHP and the proposed HKBS technique by the authors and it was proved that HKBS method for the evaluation and selection of the software components is relatively better than AHP and WSM techniques where the (i) computational efficiency (ii) ease in problem solving (iii) knowledge reuse and (iv) consistency and presentation of the evaluation results is concerned.

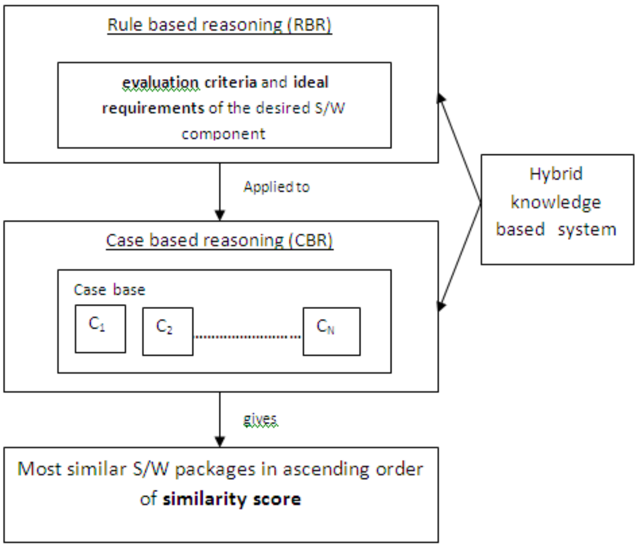

Fig 2 : working of an HKBS System

IV. Conclusion & Future Work

This paper has focused on the various formal software component selection techniques. The main objective of this paper is to evaluate the short comings of the various existing techniques of COTS component selection, be it manual, semi-automated or automated. It has been found that each technique has its own benefits and limitations; no technique is best for every case. The automated techniques are obviously better than the manual ones because they provide us with a higher accuracy of correctly selecting the desired component out of a wider pool of prospective candidate components. However, the semi-automated technique proposed [7] only narrows down the choices for selection and does not give the best solution of out the alternatives. The future scope could be to establish a fully-automated technique, which can use the expertise of the semi-automated technique proposed, to help the user select the best component according to his needs out the various alternatives.

References

[1] I. Sommerville, Software Engineering, International computer science series, 7th Edition, Addison Wesley Longman, 2005

[2] 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.

[3] 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

[4] 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.

[5] 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

[6] E. Triantaphyllou, Multi-Criteria Decision Making Methods: A Comparative Study, Kluwer Academic Publishers, 2001.

[7] V. Maxville, J. Armarego, and C.P. Lam, “Intelligent Component Selection”, proceedings of 28th annual international computer society and application conference, COMPSAC, 2004.

[8] 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.

[9] 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.

[10] 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.