A Retrospection of Software Component Selection Techniques using Computational Intelligence

Jagdeep Kaur
Department of Computer Science & Engineering/ Information Technology
The NorthCap University,Gurgaon, Haryana, India

Pradeep Tomar
Department of Computer Science & Engineering, School of ICT,
Gautam Budhha University, Greater Noida, Uttar Pardesh, India

Abstract- In this study, a capacious survey of existing component selection techniques is done. A component is selected based on the functions it provides along with some other extra functional properties. There may be many candidate components for providing the same functionality. Many techniques have been proposed for component selection under varied situations. The main emphasis is on the usage of computational intelligent techniques in component selection. One approach to combine the existing fuzzy clustering techniques, fuzzy c-means and subtractive clustering, is proposed. After analyzing the techniques it was observed that it suffers from major demerits like specifying the number of clusters beforehand and generating clusters on the basis of limited number of features. A new methodology based on fuzzy clustering is proposed here. It will work on the basis of fuzzy relations instead of similarity measures. The proposed methodology is to be be applied on a case study for further validation.

Keywords – Component Selection, COTS, Component Selection Problem, Fuzzy Clustering, Fuzzy c-means, Subtractive Clustering, Component Based Software development, Component Based Development

I. INTRODUCTION

Component-based Software Development promotes acquisition, adaptation and integration of reusable software components to swiftly develop and deploy complex software system with least engineering effort and resource cost. Here the notion of building a software by writing code has been replaced by selecting, assembling and integrating software components (*Gill,2003*). The building of systems from components and the building of components for different systems requires established methodologies and processes not only in relation to the development/maintenance aspects, but also to the entire component and various aspects of system lifecycle *(Crnkovic, 2002)*. The key point of component based software engineering is to reuse already developed software code to develop cost effective software, which requires less development time and reduces time-to-market, without compromising on the quality of the software. Reuse of the time tested software component is likely to lead to better productivity and quality. Components for development of software products can be selected from in-house developed component repository or could be purchased from commercial-off-the-shelf (COTS) vendors. According to (*Clements, 1996)*, CBSE embodies the "the 'buy, don't build' philosophy". Component selection is one of the most crucial steps in component based software development and success of the final system largely depends on the component selection process. The software selection process helps in searching the components that provides desired functionality, from the finite set of component set. The selection process helps in selection of the optimal set of components from the third party repository, on the basis of given set of requirement or constraints.

II. COMPONENT SELECTION PROBLEMS

A component is an independent executable entity that can be made up of one or more executable objects. In a component selection problem the application developer For a component selection problem the application developer considers several components based on the available information like the cost of the components, functionality of the components, time required to assemble and develop the system, performance, reliability, maintainability of the components as well as the developed system. The application developer may incorporate many other factors for making decision about the selection process like the dependencies between components, the

priority assigned to each of the customers. From the available dataset of all components, the application developer search for those components that gives a right mix of the desired features. The search space is very huge and complex as the number of candidate components is very large. As a result it becomes difficult for the application developer to find optimal subset of components that satisfies all the constraints without any programmed assistance . The component selection helps the application developer to decide which component combinations will make best choices. Component selection is playing main role in CBSE. According to *Vescan (Vescan,2009)* there are two types of component selection problem

Simple Component Selection Problem (SCSP)
Simple Component Selection Problem (SCSP) is the problem of picking a number of components from a large set of components in such a way that the resultant system is in accordance to stated requirements. This problem assumes that all the stated requirements can be fulfilled from the available set of components.

Criteria-Based Component Selection Problem (CCSP)
Similar to the SCSP, the Criteria Component Selection Problem (CCSP) is concerned with creating such a composed system which is not only satisfying the stated requirements but also various criteria are satisfied.

Now, the authors present a brief survey of software component selection techniques using computational intelligent.

III. RETROSPECTION OF SOFTWARE COMPONENT SELECTION TECHNIQUES BASED ON COMPUTATIONAL INTELLIGENCE

As the number of software components available are quite large selecting a suitable set of components that satisfy set of requirements while reducing the cost is very difficult. This problem dictates the need of design of algorithm to solve the component selection problem. This section presents the review of software component selection techniques based on various computational intelligent techniques like genetic algorithm, fuzzy logic, fuzzy clustering, rule based and case based reasoning, quality function deployment etc.

According to *Nakkrasae S et al.* (*Nakkrasae S et al., 2004*) the components in software component repository are classified into similar component cluster group with the help of Fuzzy Subtractive Clustering algorithm. The classification index is proposed based on the center of each cluster and it is compared with the required software component. The component giving the closest match is further used for component selection process.

*Stylianou and Andreou (Stylianou and Andreou, 2007),*proposed a hybrid software component clustering and retrieval scheme using an entropy-based fuzzy k-modes algorithm. It employs an entropy-based fuzzy *k*-modes clustering Algorithm, which has two main purposes. The first task is to calculate the number of clusters from the given dataset. The second task is to partition the software components into clusters with degrees of membership and find the final cluster centers. The techniques stops when it finds the cluster centers closer to user's search preferences.

In the field of mobile computing, a component selection framework is proposed by *Dong et al. (Dong et al., 2009).* It works on the principles of case-based reasoning and helps in component selection. Two integeral parts of the component selection process are: context awareness and personalization. The main advantage of this approach is it reduces the maximum memory consumption and improves the reasoning accuracy.

According to *Serban (Serban et al., 2009)* component selection process is solved using metrics and some fuzzy clustering algorithm. Initially components are selected based on some quality attributes of the components. Some metrics are defined to quantify the features of components considered for selection. The metrics used are cost, provided service utilization, required services utilization and functionality. The components are considered as objects to be clustered and by using the fuzzy clustering algorithm appropriate clusters are formed.

According to the *Seth et al. (Seth et al., 2009),* a fuzzy rule based model can be used for estimating the efforts in selecting components for application development. The factors considered for selection includes *reusability, portability, functionality, security and performance.*

According to *Kwang et al. (Kwang, Mu J, Tang and Luo, 2010),* by using genetic algorithm the optimal selection of software components is done based on minimum coupling and maximum cohesion of software modules. In this technique those software components are picked up where the cohesion of software component within a software module are maximized and coupling of software components among software modules are minimized.

According to *Sen et al.(Sen et al.,2010)* view software component selection process is mostly based on functional requirements and non-functional requirements are ignored.
Here, the functional and non-functional requirements driven fuzzy quality function deployment approach is used for software selection. The main parts of this approach are the four phases *1) Product planning 2) Product design 3) Process planning 4) Process control*. This study focuses on only first phase; it can be extended for other phases. Some other important points that came into light are: The approach needs to be modified for large number and changing requirements. The trade-offs and interdependencies among the requirements should be considered and requirement prioritization techniques can be used. Further verification is required to ensure validation of input data. For preparing the non-functional criteria template software experts can be involved to identify important and unimportant factors.

Serban et al. (*Serban et al., 2010*), presents a formal approach for component based assessment. The main components of this model are: the assessment domain, objectives, formal definitions of metrics and measurement result analysis method. Here system entities are identified, and then properties of these entities are identified using formal specifications. Lastly a problem is identified while interpreting the assessment results so fuzzy clustering analysis is proposed to place a component in more than one cluster and hence reducing the rigidity of threshold values of metrics. This model is scalable and general as other properties and interactions can be added easily.

Another technique proposed by *Kwong* (*Kwong et al,2010)* is based on Genetic Algorithms. Here the selection of software components is done on the basis of minimum coupling among the components and maximum cohesion between them, meanwhile the functionality should be as per desired features mentioned by the application developers. The maximum cohesion will result in higher reusability and minimum coupling will lead to better maintainability of software conponents.

According to *Upadhyay, Deshpande and Agrawal* (*Upadhyay, Deshpande and Agrawal, 2011)* most of the existing component selection techniques are concentrated on functional properties and even the non functional properties are for end user point of view only. From the system designer, component selector, component acquirer and system integrator point of view very less research has been done. The methodology consists of digraph and matrix approach for interaction of sub-characteristics. An index has been proposed to rank the components. Based on this user can evaluate and rank potential candidates in component design and development.
Tang,(Tang et al,2011) proposed optimization model for component selection for multiple applications. The objective of this model is to minimize all types of cost involved, viz. adaption cost and procurement cost but also considers compatibility among components

According to *Jha (Jha P.C. et al., 2011),* the fuzzy approach can be used for component selection using "Build-or-Buy" strategy in designing software structure. The selection is done on the basis of cost and non-functional factor reliability. Two fuzzy multi-objective optimization models have been proposed to solve the component selection problem. The first model maximizes the system reliability and minimizes the cost. The second optimization model considers compatibility between different alternatives of the module.

According to *Jadhav and Sonar* (*Jadhav and Sonar, 2011*) a new Hybrid Knowledge Based System (HKBS) is proposed to help decision makers in the selection process of the desired software. The basic foundation of HKBS approach is on rule based and case based reasoning techniques. The rule based reasoning, a deductive reasoning, is similar to the way human solve a problem. Whereas, case based reasoning, an inductive reasoning approach, in which problem is solved by finding solution of more similar previous cases.

According to *Tang et al. (Tang et al., 2011)* consider software component selection under multi-application development at a time. This process is using the optimization model based on nonlinear binary integer programming. As it is associated with combinatorial explosion genetic algorithm is used to solve the problem. The main assumptions are that all the enterprise applications can be developed using COTS & in house developed

components can be considered in future work. Another assumption is that that required components will be fulfilled by selecting a single component from the available component. In real scenarios required components may be fulfilled by composing available components.

Component selection process as stated by *Vescan et. al (Vescan et al.,2011)* is solved by an evolutionary algorithm with four objectives. The main feature of this methodology is the use of a repair mechanism where the previous solution of similar situation is used for the selection process. The component composition is considered as static and component selection is considered in a dynamic or changing environment with two different cases: the changing requirements and the availability of the component changes over time. When component selection includes non-functional properties like performance, reliability etc. such dependency will arise. This approach can be extended for multilevel component composition.

For improving component selection process a decision support system based on cross referencing and multiple criteria decision is proposed by *Becker (Becker et al.,2013).* The authors presented a methodology to analyze the decision criteria in such a way that number of impact factors like criterion selectivity, range, significance etc can be used for improving decision making in the similar scenarios. The complete methodology is demonstrated on the set of 14 real decision making case studies.

Here, in this section the application of computational intelligent technique in software component selection algorithms is presented. The main advantage of using such techniques is that the rigidity introduced due to use of some pre-defined metrics can be eliminated by using computational intelligent techniques. Moreover, the dependency on human judgment, which may vary, according to the expert's experience, can also be eliminated.

IV. MOTIVATION FOR USING FUZZY CLUSTERING IN COMPONENT SELECTION

The software component selection problem can be solved by using fuzzy clustering techniques. As the components have different attributes and the final components are the most optimal out of the available component sets. The usage of the existing fuzzy clustering techniques viz. fuzzy c-means and subtractive clustering generates fuzzy clusters with respect to two features only at a time only. The methodology that guides the usage of appropriate fuzzy clustering strategy is shown in Fig.1.

- The reduction in search space for component retrieval and clustering based on multi-attributes rather than considering one or two attributes, as compared to built-in functions of Fuzzy c-means and subtractive clustering Matlab (Bataineh et al., 2011).
- The use of fuzzy clustering based on fuzzy relations has eased the task of selection as the tedious task of minimization or maximization of objective functions is eliminated. Even the distance functions can not be applied for subjective type dataset.
- There is a dependency of clustering results on the distance metric for similarity or dissimilarity.
- The simple fuzzy clustering technique requires the need of mentioning the number of cluster centres beforehand, in case of Fuzzy c-means and the radii of the cluster in case of subtractive clustering. This was demonstrated by (Bataineh et al., 2011).

Due to the above mentioned reasons, one way of component selection can be the use of either fuzzy c-means or subtractive clustering. This is represented in *Fig. 1.* Here, the process starts with determination of factors affecting software component selection process. After creating the database of the factors, some degree of importance is assigned to these factors. After examining the suitability to use the type of clustering technique, next move consists of either applying Fuzzy c means or Subtractive Clustering. After the application of clustering technique and generation of the clusters, an inference system is designed.

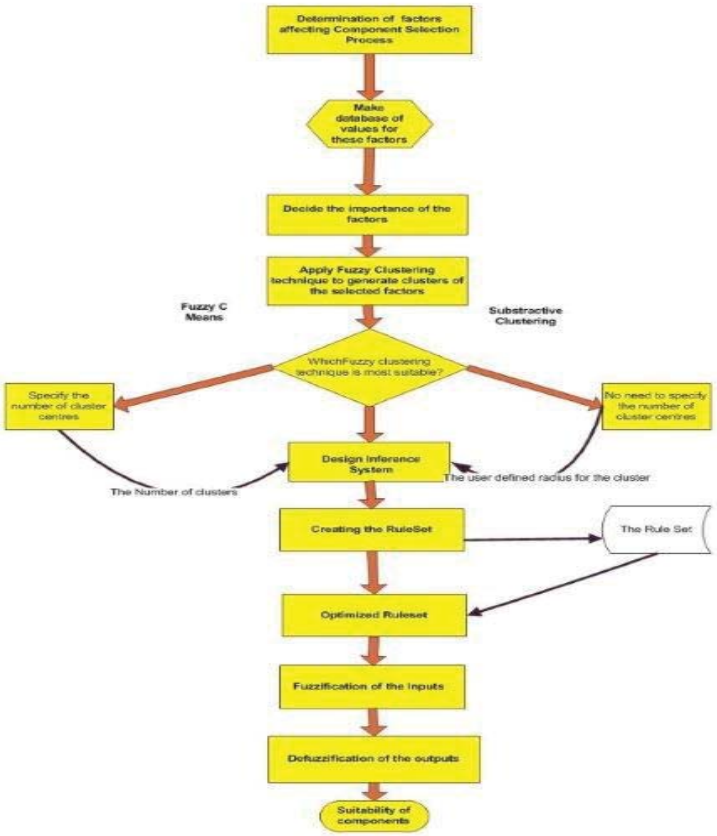

Figure 1. A combined methodology to use Fuzzy clustering Techniques

Here, the rule set created is optimized and subsequent fuzzification and defuzzification of the inputs and outputs respectively is done. Finally the suitability of the components. But due to the reason that it considers only two features at a time and other limitations mentioned above, it is more suitable to use fuzzy clustering as it deals with multi criteria solutions. This proposed approach is explained in the next section.

V. PROPOSED ALGORITHM

The main steps of the algorithm are shown in *Fig. 2*. In this algorithm, the input is set of n software components along with their features, this set is called dataset. In the first step, the dataset is normalized by taking the ratio of common features to total number of features. In the second step, take the fuzzy transitive closure of the normalized matrix.

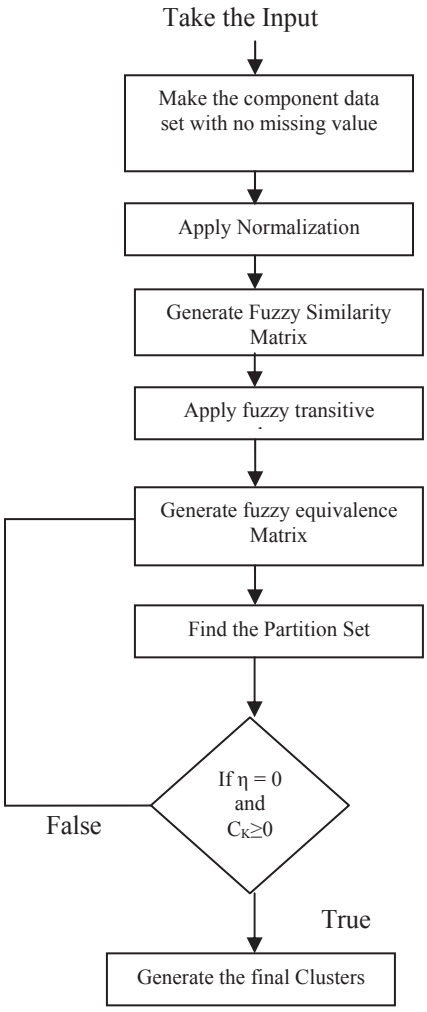

Fig. 2 Flowchart of the algorithm

For ease of computation max-min composition is used with the following formula. Using the Max-Min composition

$\mu_s(P,R) = \text{Max}_{y \varepsilon Y}(\text{Min}(\mu_R(P,Q), \mu_Q(Q,R)))$

$A^{(0)} = \Delta$

$A^{(1)} = A^{(0)} o A^{(0)}$

$A^{(2)} = A^{(1)} o A^{(0)}$

In the third step, the partitions are generated based set similar to property. In fourth step, clusters are formed. From fifth steps onwards the termination condition is checked and clustering continues till all the components are not assigned any cluster. Finally, the clusters are generated along with their membership values.

V.CONCLUSION

This paper discusses the software component selection problem. There are various ways to deal with it, one is by using computational intelligent techniques like genetic algorithm, fuzzy logic, fuzzy clustering, rule based and case based reasoning, quality function deployment etc. This paper surveys the software component selection techniques using these intelligent techniques. The two main contributions of this paper are: firstly, a combined approach of fuzzy clustering technique for component selection and secondly, a fuzzy relation-based approach for fuzzy clustering of software component selection is proposed. The future work will be based on validating these approaches on a suitable case study.

REFERENCES

[1] Becker C., Kraxner M.,Plangg M. and Andreas Rauber "Improving decision support for software component selection through systematic cross-referencing and analysis of multiple decision criteria ",HICSS 2013 Multi-criteria Decision Support, January 2013.

[2] Crnkovic, I. (2002), "Component ‐ Based Software Engineering-New Challenges in Software Development". Software Focus, Vol. 2, No. 4, *pp. 127-133*.

[3] Clements, P. C. (1996), "From Subroutines to Subsystems: Component-Based Software Development". American Programmer, Vol. 8, *pp. 31-31*.

[4] Dong F., Zhang L., Hu D. H., and Wang C. L "A Case-Based Component Selection Framework for Mobile Context-Aware Applications", in *proc. of International Symposium on Parallel and Distributed Processing with Applications,* 2009, pp.366-373.

[5] Gill, N. S., (2003), "Reusability Issues in Component-Based Development", ACM SIGSOFT Software Engineering Notes Vol. 28 No. 6, *pp. 30*.

[6] Hamza-lup, G. L., Agarwal, A., Shankar, R., and Iskander, C. (2008). Component Selection Strategies Based on System Requirements Dependencies on Component Attributes,proceedings of *IEEE International Systems conference, pp.1-5.*

[7] Jha P. C., Arora R. and Kumar U. D., "A Fuzzy Approach for Components Selection amongst Different Versions of Alternatives for a Fault Tolerant Modular Software System under Recovery Block Scheme Incorporating Build –or-Buy Strategy", vol. 1, no.4, *American Journal of Operation Research*, pp. 249-258, 2011.

[8] Jadhav, A. S., and Sonar, R. M. (2011). Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach.,*Journal of Systems and Software*, Vol.84 No.8, *pp.1394-1407.*

[9] Kwang C. K ,Mu L F,Tang, J F and Luo X G,"Optimization of Software Component Selection for Component-Based Software System Development", Journal of Computers & Industrial Engineering, Elsevier, pp 618-624,2010.

[10] Kaur, A. and Mann K. S., (2010). Component Selection for Component- Based Software Engineering, *International Journal of Computer Applications*, Vol.2-No.1, *pp. 109-114.*

[11] Nakkrasae S., Sophatsathit P. and Edwards W. R. "Fuzzy Subtractive Clustering based Indexing Approach for Software Components Classification", *International Journal of Computer and Information Science* ,vol.5, no. 1, pp. 63-72, 2004.

[12] Stylianou C. and Andreou A. S., "A Hybrid Software Component Clustering and Retrieval Scheme using an Entropy-based Fuzzy k-modes Algorithm", in *proc. 19th IEEE International Conference on Tools with Artificial Intelligence, 2007*, pp. 202-209.

[13] Serban C.,Vescan A. and Pop H.F., " A New Component Selection Algorithm Based on Metrics and Fuzzy Clustering", *Creative Mathematics and Informatics*, vol. 1, no. 3, pp. 505-510, 2009.

[14] Seth, K., Sharma, A., & Seth, A. (2009). Component Selection Efforts Estimation–a Fuzzy Logic Based Approach. *International Journal of Computer Science and Security,(IJCSS), 3*(3), 210-215.

[15] Şen, C. G. and H. Baraçlı 0 0. Fuzzy Quality Function Deployment based Methodology for Acquiring Enterprise Software Selection Requirements. Expert Systems with Applications, Vol. 37, No.4 pp. 3415-3426.

[16] Tang, J. F., Mu, L. F., Kwong, C. K., and Luo, X. G ,"An optimization model for software component selection under multiple applications development, "*European Journal of Operation Research,* Vol. 212, *pp. 301-311.* 2011.

[17] Upadhyay, N., Deshpande, B. M., and Agrawal, V. P. (2011). Concurrent usability evaluation and design of software component: a digraph and matrix approach. *IET Software*,Vol. 5 Issue 2, *pp.188-200*.

[18] Vescan, A. (2008). Pareto dominance - based approach for the Component Selection Problem, proceedings of *Computer Modelling and Simulation EMS 08,pp.58-63*.

[19] Vescan A., Grosan C., Shengxiang. Y. "A hybrid evolutionary multiobjective approach for the dynamic component selection problem", in Proc. of *11th International Conference on Hybrid Intelligent Systems (HIS)*, 2011, pp. 714-72.