Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018)
IEEE Conference Record: # 42666; IEEE Xplore ISBN:978-1-5386-3570-4

A NOVEL SOFTWARE COMPONENT SELECTION THROUGH STATISTICAL MODELS

K.R.Sekar, J.Sethuraman, R.Manikandan

School of Computing, SASTRA University, India

ABSTRACT----Software components are the inevitable commodity for all web pages to provide bountiful services. The objective of the reusable Software Component couplings and cohesions helps in providing big services to the web pages and acting as a backbone. In the study 82% of the commercial components play a vital role and remaining percentage are considered here as customized components. Components interoperability is possible via intra-inter components. Customized components are economical in nature and the above said components can be utilized for the empirical applications. Selection towards components and identifying software components for design needs extraordinary methodologies and techniques employed like Classifications, Clustering and Associations provide a greater mileage and milestone for above mentioned methodology. The applied methodologies were Rank order for clustering; k-means clustering and similarity based coefficients. The result of the study shows the selecting the best software components for web applications. The research work concludes reusable software components are providing better services to the web applications at a affordable price.

Keywords: Commercial components, Customized components, Classification, Clusters, K-Means, Rank order methodology and Similarity coefficient ranking.

I.INTRODUCTION

Components are the ingredients of applications which are put together to form the mentioned job to be done. The proposed system contains software components which are the interfaces called to provide the functionality that the component capsulated. Software components have the properties which are homogeneity and heterogeneity. The is the direction to achieve software components are component selection and component identification. There are different methods for identifying components like graphical representation and clustering methodologies. Scheduling the time factor for producing such a type of components is difficult and sometimes it may take more time. Using semantic web plenty of services can be in the web applications [1]. Selection of the reusable components are made via decision model. The optimization can be made using lower the cost and increases the service and reliability [2]. In this work mainly focused on quality of software systems. Reliability is the important factor to be considered for arranging clusters that directly proportional to the cost and services [3].

This proposed system has methodologies K-means clustering, Rank order clustering and Similarity based coefficient. Clustering is an algorithm that clusters the components based on their attributes. In k-means clustering, shows the number of clusters were specified for the attributes to be grouped them into. This methodology assigns each attribute to a cluster and finds the centroid of each cluster then iterates through reassigning attribute points to the cluster whose centroid is closest and calculates new centroid of every cluster and the cluster cannot be reduced further. Weights are calculated through 0's and 1's. It Provides multiple components for a CBS, whenever used to accommodate services to the web pages [4]. Rows and columns values are multiplied by the binary numbers with their required position powers. Attributes are arranged in a linguistics methods got from experts in the same domains. Maximum amount dense 1's were clustered that provides the required components and their services in terms with QoS [5]. Second, relation between the QoS and the components through regression analysis. and attributes for services such as similarity based coefficient algorithm. This algorithm gives a binary product-machine for rank-order clustering of n-by-m matrix. In this work mainly focus on CBS goals were achieved through dependency analysis. And Introducing the new notion of intention dependency graph. Here, the identification of the gap is defined. It aims to investigate the state-of-the-art of the CBSE. This paper addresses five dimensions of CBSE. The main objectives found were to increase productivity, save costs and improve quality [6].

II.LITERATURE SURVEY

In this work optimization was carried out through adaptive monitoring system is used . Resources were modeled through contracts and requirements. Here there are two use cases used for different domains to evaluate the system [7].In this work the new model is used which is internal QoS for selecting OSS. The OSS systems used eight different models for finding the optimal solution for selection of useable components [8]. Different ranking scores were applied MCDM techniques used to find the ranking scores for integration of QoS with components [9]. Reverse engineering and structured models are required for comparison and inspect the work flow and gauge the reliability of the activities [10]. Heuristic and meta heuristic models are needed to find the reliability and accuracy of the work. Uncertainty will be reduced through the above said methods [11]. For measuring the metrics of the reusable components 14 proposals were available for acceptance. Quality assessment and quality measures for the CBSS is difficult that was primarily detected by the studies [12]. Analytical method is the right suited method rather the probabilistic model for real optimization for cost factor. Case study shows 28% speed-up in availability-cost case over pure evolutionary optimization [13].In this work the cycle metrics is to classify components into cyclic and non-cyclic groups. Here, the components in cyclic dependencies are more defect-prone. The defect predictions possible through refactoring the models and revised ‘N’ number of times [14]. The paper aims at proposing a data mining framework which will help in selection of suitable site for wind turbine′s installation [15].

III. PROPOSED WORKS

This figure represents the Tech Gallery application and the flow of entire process that is going to take part in article. The attributes will be given as input from the Tech Gallery, then the components will respond accordingly forming clusters and providing percentage of dominant factors which will be generated as graph.

TABLE 1. COMPONENTS AND RELATED ATTRIBUTES

Components Used	**Attributes**
Swd – Sliderwd	**Fnty -** Functionality
Ig– Ignite Gallery	**Spt -** Support
2jg – 2j Gallery	**Eou -** Ease Of Use
Rsmg– Rsmediagallery	**Oal -** Overall
Gp- Gallerypro	**Rtg –** Rating
Sg –Sim Gallery	**Upl -** Upload
Pg –Phoca Gallery	**Vfm -** Value For Money
Ssfx - Slideshowfx	**Cmn –** Customization
Cis –Creativeimageslider	**Dwld -** Download
2jg –2j Gallery	**Atpy -** Autoplay

A. Rank order clustering

Rank order clustering is an algorithm based on incident matrix and rows and columns are arranged which is given below, Continue steps until there all columns are arranged in the decreasing order.

TABLE 2. DATASET

	UPL	**FNTY**	**RTG**	**SPT**	**EOU**	**OAL**	**VFM**	**CMN**	**DWLD**	**ATPY**
SSFX	1	1	1	1	1	1	0	0	1	0
RSMG	0	1	0	1	1	1	1	1	0	0
CIS	0	1	0	1	1	1	0	1	0	1
SG	1	1	1	1	1	1	1	1	1	0
PG	0	1	1	1	1	1	1	1	0	1
LS	1	1	1	1	1	1	1	1	0	0
SWD	1	1	1	1	1	1	1	1	0	0
GP	1	1	1	1	1	1	1	0	1	0
IG	1	1	1	1	1	1	1	1	1	1
2JG	1	1	1	1	1	1	1	1	0	0

TABLE 3. ROW SORTING

	UPL	FNTY	RTG	SPT	EOU	OAL	VFM	CMN	DWLD	ATPY
SWD	1	1	1	1	1	1	1	1	1	1
IG	1	1	1	1	1	1	1	1	1	0
2JG	1	1	1	1	1	1	1	1	0	0
RSMG	1	1	1	1	1	1	1	1	0	0
GP	1	1	1	1	1	1	1	1	0	0
SG	1	1	1	1	1	1	1	0	1	0
PG	1	1	1	1	1	1	0	0	1	0
LS	0	1	1	1	1	1	1	1	0	1
SSFX	0	1	0	1	1	1	1	1	0	0
CIS	0	1	0	1	1	1	0	1	0	1

Step 1: Initially, assign binary weight.

Step 2: Calculate the weights are measured through rows by adhered formula

$$\text{Decimal weight for row} \quad i = \sum_{p=1}^{m} b_{ip} 2^{m-p} \tag{1}$$

Step 3: The rows were rearranged via decreasing order of their decimal weights.

Step 4: Continue steps until there all rows are ordered in decreasing manner.

Step 5: Calculate the decimal weight using the formula for the columns.

$$\text{Decimal weight for column } j = \sum_{p=1}^{n} b_{pj} 2^{n-p} \tag{2}$$

Step 6: Rank the columns in the order of decreasing decimal weights

TABLE 4. COLUMN SORTING FOR RANK ORDER

	FNTY	SPT	EOU	OAL	RTG	UPL	VFM	CMN	DWLD	ATPY
SWD	1	1	1	1	1	1	1	1	1	1
IG	1	1	1	1	1	1	1	1	1	0
2JG	1	1	1	1	1	1	1	1	0	0
RSMG	1	1	1	1	1	1	1	1	0	0
GP	1	1	1	1	1	1	1	1	0	0
SG	1	1	1	1	1	1	1	0	1	0
PG	1	1	1	1	1	1	0	0	1	0
LS	1	1	1	1	1	1	1	1	0	1
SSFX	1	1	1	1	0	0	1	1	0	0
CIS	1	1	1	1	0	0	0	1	0	1

B. K-Means clustering

Input:

Functionality, Support, Ease of use, Overall, Rating, Upload, Value for money, Customization, Download, Autoplay (is a set of data points as 'p').

Input is the no of clusters to be grouped.

Output gives the set of cluster which is formed.

Step1: In the Dataset, the attributes are selected for the most extreme in the column and the base component for the column.

Step2: From the most extreme extent the section is recognized.

Step3: The whole dataset is sorted by expanding the request which having the greatest extent.

Step4: The sorted dataset are set into 'c' equivalent amount of values.

Step5: The arithmetic mean is calculated for every part which is available for the calculation of centroids.

Step6: From the mean we calculate nearest group from the located nearest centroid.

Step7: Repeat for every data point.

1) The immediate nearest cluster for this centroid, we measure distance.
2) If present closest distance is not compared to the distance measured, the value remains in the cluster else,
    a) For each centroid calculated, we perform separation.
    b) Value for the data point is assigned to the nearest centroid which contains cluster. Step 8: Generate final result of clustered data.

TABLE 5 K-MEANS CLUSTERING

	FNTY	SPT	EOU	OAL	RTG	UPL	VFM	CMN	DWLD	ATPY
SWD	1	1	1	1	1	1	1	1	1	1
IG	1	1	1	1	1	1	1	1	1	0
2JG	1	1	1	1	1	1	1	1	0	0
RSMG	1	1	1	1	1	1	1	1	0	0
GP	1	1	1	1	1	1	1	1	0	0
SG	1	1	1	1	1	1	1	0	1	0
PG	1	1	1	1	1	1	0	0	1	0
LS	1	1	1	1	1	1	1	1	0	1
SSFX	1	1	1	1	0	0	1	1	0	0
CIS	1	1	1	1	0	0	0	1	0	1

C. Similarity based coefficient

When a n x m matrix of the machine learning binary product is given, the algorithm follows these steps:

TABLE 6. RESULT FOR SIMILARITY BASED COEFFICIENT METHOD

	FNTY	SPT	EOU	OAL	RTG	UPL	VFM	CMN	DWLD	ATPY
SWD	--	9/10	4/5	4/5	4/5	4/5	7/10	4/5	3/5	3/5
IG	--	--	8/9	8/9	8/9	8/9	7/9	7/10	2/3	½
2JG	--	--	--	1	1	7/9	2/3	7/9	3/4	5/9
RSMG	--	--	--	--	1	7/9	2/3	7/9	3/4	5/9
GP	--	--	--	--	--	7/9	2/3	7/9	3/4	5/9
SG	--	--	--	--	--	--	7/8	3/5	5/9	2/5
PG	--	--	--	--	--	--	--	1/2	4/9	4/9
LS	--	--	--	--	--	--	--	--	3/4	¾
SSFX	--	--	--	--	--	--	--	--	--	5/7
CIS	--	--	--	--	--	--	--	--	--	--

Step 1: Compute the similarity based coefficient using the formula

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}.$$

Step 2: Find the higher similarity coefficient with k which is the iteration index and group together in cell k the tuple (i*, j*).

TABLE 7. REDUCED SIMILARITY BASED COEFFICIENT

	(FNTY,OAL)	SPT	EOU	RTG	UPL	VFM	CMN	DWLD	ATPY
(SWD,RSMG)	--	9/10	4/5	1	4/5	7/10	4/5	¾	3/5
IG	--	--	8/9	8/9	8/9	7/9	7/10	2/3	½
2JG	--	--	--	1	7/9	2/3	7/9	3/4	5/9
GP	--	--	--	--	7/9	2/3	7/9	3/4	5/9
SG	--	--	--	--	--	7/8	3/5	5/9	2/5
PG	--	--	--	--	--	--	1/2	4/9	4/9
LS	--	--	--	--	--	--	--	3/4	¾
SSFX	--	--	--	--	--	--	--	--	5/7

CIS	--	--	--	--	--	--	--	--	--

TABLE 8. FINAL REDUCED SIMILARITY BASED COEFFICIENT

	(A1,A2,A4,A5)	(A3,A6,A8,A9)	A7	A10
(C1,C2,C4,C5)	----	8/9	###	3/5
(C3,C6,C8,C9)	----	----	7/9	3/4
C7	----	----	----	4/9
C10	----	----	----	----

TABLE 8. FINAL CLUSTERED RESULT FORMED IN SIMILARITY COEFFICIENT

	(FNTY,SPT,OAL,RTG)	(EOU,UPL,CMN,DWLD)	VFM	ATPY
(SWD,IG,RSMG,GP)	--	8/9	7/9	3/5
(2JG,SG,LS,SSFX)	--	--	7/9	¾
PG	--	--	--	4/9
CIS	--	--	--	--

Step 3: Eliminate row i* and n j* from the binary matrix.

Step 4: Substitute the values for rows and columns of the cell k with s_{rk}=max(s_{ri*},s_{rj*}).

Again the iteration index k is raised by one and proceed till all machines are kept in one single group.

TABLE 9. COMPONENTS VS DOMINANT FACTORS

COMPONENTS	DOMINANT FACTORS	PERCENTAGE
SLIDERWD, IGNITE GALLERY 2J GALLERY, RSMEDIAGALLERY GALLERY PRO,SIM GALLERY PHOCA GALLERY,LAYER SLIDER,SLIDESHOWFX CREATIVEIMAGESLIDER	FUNCTIONALITY,SUPPORT EASE OF USE,OVERALL	100%
SLIDERWD, IGNITE GALLERY 2J GALLERY, RSMEDIAGALLERY GALLERY PRO,SIM GALLERY PHOCA GALLERY,LAYER SLIDER	RATING,UPLOAD	93%
SLIDERWD,IGNITE GALLERY 2J GALLERY, RSMEDIAGALLERY GALLERY PRO,SIM GALLERY PHOCA GALLERY, LAYER SLIDER, SLIDESHOWFX CREATIVEIMAGESLIDER	VALUE FOR MONEY CUSTOMIZATION	80%
SLIDERWD,IGNITE GALLERY	DOWNLOAD,AUTOPLAY	**75%**

Table 10 Final Result for calculated percentage formed for attributes and components

COMPONENT 1, COMPONENT 2 COMPONENT 4, COMPONENT 5	EASE OF USE,UPLOAD,VALUE FOR MONEY CUSTOMIZATION,DOWNLOAD,AUTOPLAY
COMPONENT 3, COMPONENT 6 COMPONENT 8, COMPONENT 9	VALUE FOR MONEY,DOWNLOAD AUTOPLAY
COMPONENT 7	AUTOPLAY

IV. RESULTS AND DISCUSSIONS

In Table-2 Tech gallary software components were taken from the joomla CMS. In Table3 and Table-4 rank order coeffiecients were applied for clutering dominance attributes for selcting best software components. Table- 5 shows the K-means clusters towards the software components for features supporting tech gallaries. Table-9 exhibits the similarity coefficients for selecting resuable software components. Calculations and evolutions were tested through the clear satistical methods and obtained 91.73% results.

V.CONCLUSION

Thus, the main theme of building an application for identification of best component is achieved. In our proposed application the best component is identified depending on several attributes like overall, ease of use, customization, support, value for money. In addition to these, there were few other factors like upload, functionality, auto play, download and rating. These clusters are constructed based on the historical data available in internet. By using these clusters we can identify the best component from several numbers of components for various applications.

VI.ACKNOWLEDGEMENT

I really appreciate efforts of Ms.Sowmya Vantala, Ms.M.Manisha Jain, Ms.P.Drakshayani students from final year 2017, School of Computing who have done a remarkable job by implementing my ideology and methodology into a real time article.

REFERENCES

1. Kaur, L., & Mishra, A. , Software component and the semantic Web: An in-depth content analysis and integration history. Journal of Systems and Software, 2017, 125, 152-169.

2. Zhiqiao, W., Kwong, C. K., Tang, J., & Chan, J. W. K., Integrated model for software component selection with simultaneous consideration of implementation and verification. Computers & Operations Research,2012, 39(12), 3376-3393.

3. Pham, T. T., Défago, X., & Huynh, Q. T. , Reliability prediction for component-based software systems: Dealing with concurrent and propagating errors. Science of Computer Programming, 2015, 97, 426-457.

4. Khan, M. A., & Mahmood, S., A graph based requirements clustering approach for component selection. Advances in Engineering Software, 2012, 54, 1-16.

5. Jha, P. C., Bali, V., Narula, S., & Kalra, M., Optimal component selection based on cohesion & coupling for component based software system under build-or-buy scheme. Journal of Computational Science,2014, 5(2), 233-242.

6. Vale, T., Crnkovic, I., de Almeida, E. S., Neto, P. A. D. M. S., Cavalcanti, Y. C., & de Lemos Meira, S. R., Twenty-eight years of component-based software engineering. Journal of Systems and Software,2016, 111, 128-148.

7. Gonzalez-Herrera, I., Bourcier, J., Daubert, E., Rudametkin, W., Barais, O., Fouquet, F., ... & Baudry, B., ScapeGoat: Spotting abnormal resource usage in component-based reconfigurable software systems. Journal of Systems and Software, 2016, 122, 398-415.

8. Sarrab, M., & Rehman, O. M. H. , Empirical study of open source software selection for adoption, based on software quality characteristics. Advances in Engineering Software, 2014, 69, 1-11.

9. Zaidan, A. A., Zaidan, B. B., Hussain, M., Haiqi, A., Kiah, M. M., & Abdulnabi, M. , Multi-criteria analysis for OS-EMR software selection problem: a comparative study. Decision Support Systems, 2015, 78, 15-27.

10. Minetola, P., Iuliano, L., & Calignano, F., A customer oriented methodology for reverse engineering software selection in the computer aided inspection scenario. Computers in Industry,2015, 67, 54-71.

11. Pitangueira, A. M., Maciel, R. S. P., & Barros, M., Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature. Journal of Systems and Software,2015, 103, 267-280.

12. Abdellatief, M., Sultan, A. B. M., Ghani, A. A. A., & Jabar, M. A. , A mapping study to investigate component-based software system metrics. Journal of systems and software, 2013, 86(3), 587-603.

13. Koziolek, A., Ardagna, D., & Mirandola, R.,Hybrid multi-attribute QoS optimization in component based software systems. Journal of Systems and Software,2013, 86(10), 2542-2558.

14. Oyetoyan, T. D., Cruzes, D. S., & Conradi, R., A study of cyclic dependencies on defect profile of software components. Journal of Systems and Software, 2013, 86(12), 3162-3182.

15. Shaheen, M., & Khan, M. Z., A method of data mining for selection of site for wind turbines. Renewable and Sustainable Energy Reviews, 2016, 55, 1225-1233.