New Component Composition Metrics for Component based Software Development

Rajender Singh Chhillar
Department of Computer Science & Applications,
Maharshi Dayanand University,
Rohtak-124001, Haryana, India

Parveen Kajla
Department of Computer Science & Applications,
Maharshi Dayanand University,
Rohtak-124001, Haryana, India

ABSTRACT
Component Based Software Engineering (CBSE) is anenvironment which uses software components as main building block during designing and creation of a software system. A Component is a software entity with independent identity which has a perceptible reusable interface. This property motivates the programmer to design and develop Softwareusing Component Based Software Development (CBSD) and further Software Metrics for these systems. In this paper, new early stage component based software metrics are designed for CBSD namely Component Composition Metrics (CCM) & Component Ratio Metrics (CRM) to determine the Effort using Likert 3-point rating in terms of time, cost, quality, operability, changeability, adaptability etc.for a software system. CCM and CRM are designed and analyzed using knot model of component based software life cycle.

Keywords
Component, CBSD, KNOT Model, CCM, CRM, Effort

1. INTRODUCTION
A component is a software system or subsystem that can be factored out or broken intoan independent entity, which has a potentially reusable exposed interface [1]. Acomponent encapsulates its constituent features and hence is never deployed partially[2].
Component Based Software Development is the major approach in the software development having its factors like configurability, reusability, complexity, deployment, maintainability, scalability, testability etc. This approach leads the development effort easily accessible reducing the cost and time along with better productivity. A component plays a vital role in the composition as it may affect the whole process. Hence, addition or removal if any must be operative, leaving no space of modification for other components. Besides component, a life cycle model performs well from its inspection to retirement. It maps the different activities exercised on software in a disciplined and systematic way. Therefore, a development team must identify a suitable life cycle model and metrics proposed by different researchers in their project to enhance the efficiencies and outcomes.
And it must be taken into consideration that the metrics proposed for early stages counted as more productive and beneficiaries than the latter one. In this way, Composition Metrics attracts the attention more during the development process of CBSD. Composition Metrics broadens the scope of research in other quality factors leaving behind the other attributes of software including reusability, complexity, deployment etc.
To determine component composition and the efforts for the composition during the system development at the design stage of the CBSD a Component Composition Metric (CCM) and Component Ratio Metrics(CRM) are introduced in exploring component composition. This metric is used with the composition of five and ten components in the research and finally the results are analysed for the component based software development.

This paper is organized into seven sections. Section 2 describes the component based software development methodologies proposed by the various researchers. Section 3 defines the new proposed Component Composition Metric (CCM) and Component Ratio Metrics (CRM) for the CBSD. Section 4 analyse CCM & CRM applying Likert 3-point scale on different sets of component showing tabular & graphical representation of results.Section 5 presents the Results and observation; Section 6 refers concluding remarks and future scope. At last acknowledgements are shown in section 7.

2. COMPONENT BASED SOFTWARE DEVELOPMENT
In CBSD, numbers of process models were proposed, Luiz, 2005 [5] proposed Y component-based software life cycle model considering iteration and overlapping during its phases. The assembling and archiving is done in the iteration phases in the middle of the process. Therefore, metrics are not used during early stages. Gill N.S and Tomar P., 2008 [4] proposed X model for component-based software life cycle which represents software development for reuse and software development with or without reuse. In this model Analysis, specification and designing was done at each branch of X model. Srivastava, Chauhan, Raghuraj, 2011 [6] proposed Square Model, which is very specific for call center software application and should be developed in dedicated environment. Chhillar, Kajla [7], proposed knot model of component based software development,having capability to use existing component from reservoir and reuse it by modifying it to an extent and create new components, if required. During different phases of this model, it follows prevalent functional model like increment, prototyping and spiral model and depicts analysis, testing and feedback to reduce risk. Considering all these, Knot Model is followed to implement the proposed metrics.

3. PROPOSED COMPONENT METRICS

3.1 Component Composition Metric (CCM)
In the beginning of the system design a metric is required which results in the reusability composition of the whole system. Components are regularly composed for the purpose of offering more services in a system. This composition creates reusability among components. The existing and proposed Models of CBSD specifies that assembled components are emerged from different criteria's during their development phases. Therefore, efforts required for development of a component varies and thus each component in the system plays its role in the composition.

Assembling between the components can use different efforts to regroup them, thus efforts made for individual component must be considered during the new component based system.From the knot model it is considered that Component composition is the sum of all the components in the new system.

So Component Composition Metric (CCM) is defined as

$$CCM = \mathrm{CC}_{rc} + \mathrm{CC}_{nc} + \mathrm{CC}_{mrc}$$

where$CC_{rc}$is the sum of all the components from the Reusable Component pool, CC_{nc} is the sum of all the components which developed from the beginning and CC_{mrc} is the sum of all the components which are modified from the existing pool.

Thus CC_{rc} is defined as

$$CC_{rc} = CC_{rc_1} + CC_{rc_2} + \ldots\ldots + CC_{rc_n}$$

$$\mathrm{CC}_{rc} = \sum_{i=0}^{n} CC_{rc_i}$$

$CC_{nc}$is defined as

$$CC_{nc} = CC_{nc_1} + CC_{nc_2} + \ldots\ldots + CC_{nc_m}$$

$$\mathrm{CC}_{nc} = \sum_{j=0}^{m} CC_{nc_j}$$

And CC_{mrc} is defined as

$$CC_{mrc} = CC_{mrc_1} + CC_{mrc_2} + \ldots\ldots + CC_{mrc_p}$$

$$\mathrm{CC}_{mrc} = \sum_{k=0}^{p} CC_{mrc_k}$$

Therefore CCM is defined as

$$\mathrm{CCM} = \sum_{i=0}^{n} CC_{rc_i} + \sum_{j=0}^{m} CC_{nc_j} + \sum_{k=0}^{p} CC_{mrc_k}$$

3.2 Composition Ratio Metrics (CRM)

From the Component Composition Metric (CCM) a few new metrics are also proposed to determine the ratio of composition and these metrics are called as Composition Ratio Metrics (CRM). CRM are helpful in determining the ratio of reusable component, modified reusable component and new components required for a software development. This composition helps in determining precious factors of development like time, cost, adaptability etc. So Component Ratio Metrics (CRM) are defined as

$$CRM_{rc} = \left(\frac{CC_{rc}}{CCM}\right) * 100$$

$$CRM_{mrc} = \left(\frac{CC_{mrc}}{CCM}\right) * 100$$

$$CRM_{nc} = \left(\frac{CC_{nc}}{CCM}\right) * 100$$

4. ANALYSIS OF COMPONENT COMPOSITION IN CBSD

One of the main factors to analyse during the composition of components is the estimation of Effort required to assemble the variety of components in the system. In 1981, Barry W. Boehm developed an algorithmic software cost estimation model 'Constructive Cost Model' (COCOMO) as a model for estimating effort, cost, and schedule for software projects.[8] These projects were based on the waterfall model of software development which was the prevalent software development process during that period.
In 1995 COCOMO II was developed and is better suited for estimating modern software development projects. [9] The need for the new model came as software development technology moved from batch processing to desktop development, code reusability and the use of off-the-shelf software components.
Irrespective of Basic COCOMO and Intermediate COCOMO in detailed COCOMO, the effort is calculated as function of program size and a set of cost drivers given according to each phase of software life cycle.

But now days with new emerging trends of software development like CBSD, considering the cost driversfor each project, are difficult to determine. Component Composition Metric is analysed by giving rating to the different cases using Likert 3-point scale for Effort calculationin our study.

4.1 Effort Calculation

Efforts are made for the development of software. In our study these efforts are calculated using the Likert 3-point scale which is used as probability of the composition as

Minimum Efforts (1): Reusable Component

Average Efforts (2): Modified Reusable Component

Maximum Efforts (3) : New Component

So,

$$Eff_{rc} = 1 * nCC_{rc}$$

$$Eff_{mrc} = 2 * nCC_{mrc}$$

$$Eff_{nc} = 3 * nCC_{nc}$$

where nCC_{rc} is the total number of Reusable Component, nCC_{mrc} is the total number of Modified Reusable Components andnCC_{nc} is the total number of new components in the new system.

Then CCM effort is calculated as:

$$CCM_{eff} = Eff_{rc} + Eff_{mrc} + Eff_{nc}$$

The research includes all possible cases of 5 and 10 components to determine the effort in the CBSD, considering the number of reusable components, modified reusable component and new component in the above equations. To determine No. of Cases (NC) along with components (n):

$$NC = \frac{n * (n+1)}{2} + n + 1$$
$$= \frac{n^2 + 3 * n}{2} + 1$$

Table 1 No. of cases with n-components

Components (n)	No. of Cases (NC)
5	21
10	66
50	1326
100	5151

After then CCM Effort is calculated and Mean is determined for all cases. The result is shown in table 2 using 5 components.

Table 2Mean Table of Likert 3-pointrating scale with 5 components

NC	nCCrc	nCCmrc	nCCnc	CCM Eff	Mean
1	5	0	0	5	1
2	4	1	0	6	1.2
3	3	2	0	7	1.4
4	4	0	1	7	1.4
5	2	3	0	8	1.6
6	3	1	1	8	1.6
7	1	4	0	9	1.8
8	2	2	1	9	1.8
9	3	0	2	9	1.8
10	0	5	0	10	2
11	1	3	1	10	2
12	2	1	2	10	2
13	0	4	1	11	2.2
14	1	2	2	11	2.2
15	2	0	3	11	2.2
16	0	3	2	12	2.4
17	1	1	3	12	2.4
18	0	2	3	13	2.6
19	1	0	4	13	2.6
20	0	1	4	14	2.8
21	0	0	5	15	3

Analysing Table 2 of 5-components, which represents that in Case-1 where all the components are from reusable pool is best with Mean=1. Hence efforts are required only to assemble the components without modifying or creating new components. Practically it occasional occurs in small projects only. In Case-21 where all the components are created from initial stage is worse composition with Mean=3. It violates the rule of component-based software development. The middle of the table represents good and average results with mean value greater than 1 and less than 3.

Using Table-2, figure-1 and figure-2 shows the graphical representation with 5 components. Figure-3 shows the graphical representation of 10-components.

Figure 1 Effort-Scale (line) with 5-Components

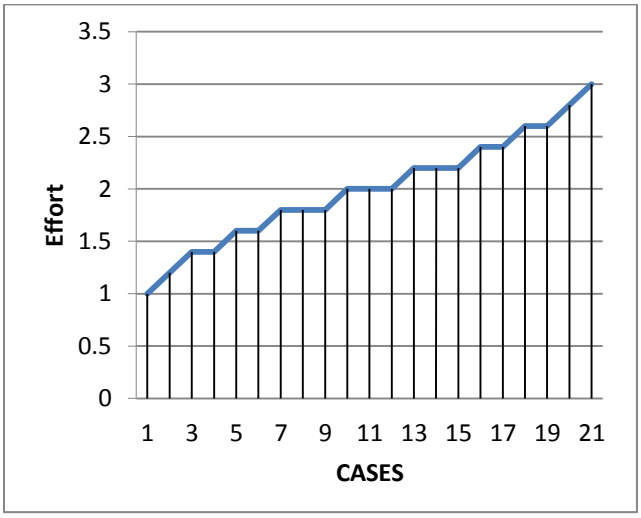


Figure 2Effort-Scales (Scatter) with 5-Components

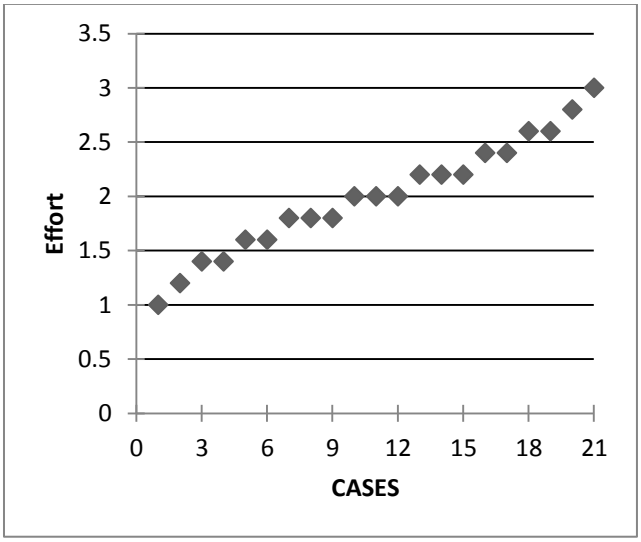
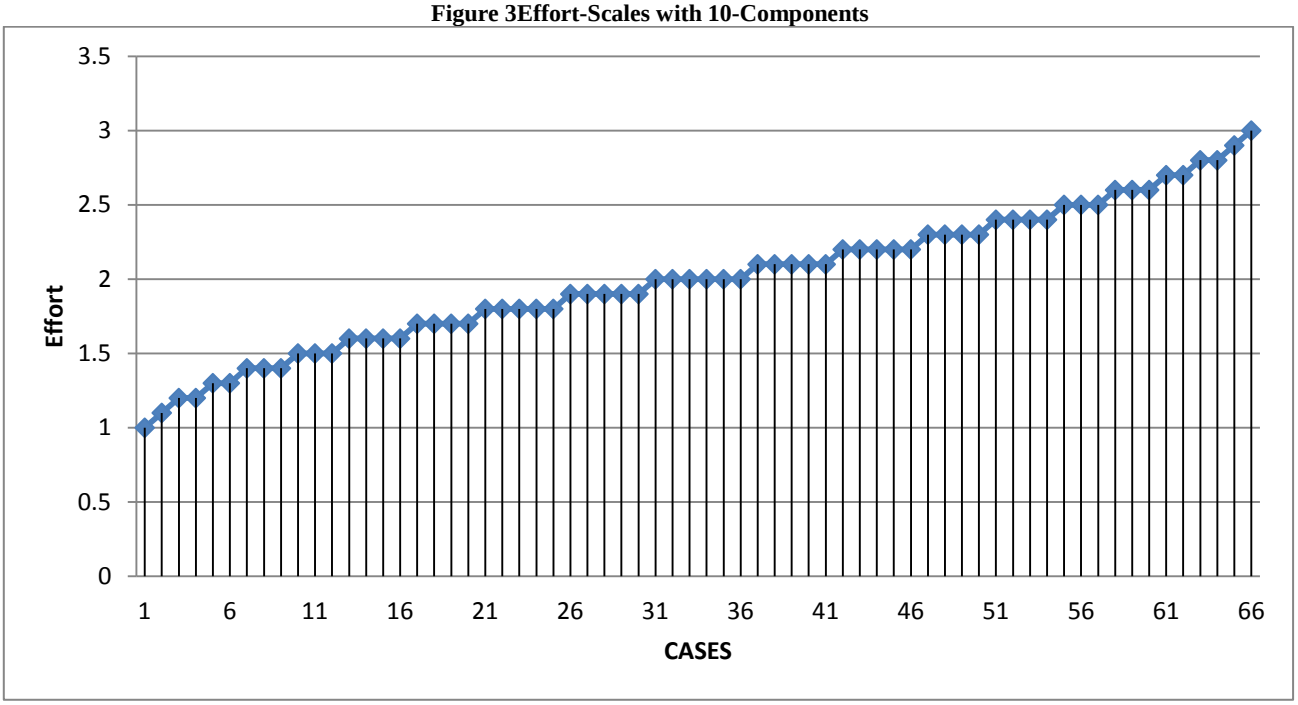


Figure 3Effort-Scales with 10-Components



Considering the above facts the composition of components iscategorized into four major categories in table-3.

Table 3 Analysis of composition and its categories

Category	range		No. of Cases	
	min	max	5 Component	10 component
Outstanding	>1.0	<= 1.5	4	12
Good	>1.5	<=2.0	8	24
Average	>2.0	<=2.5	5	21
Below Average	>2.5	<=3.0	4	9

5. RESULTS

CCM is used to determine the composition of Software Development using CBSD and thus helps in determining the efforts in the early stages of software development.CRM is used to determine the ratio of components in the system and thus indicates the percentage of components in the system.

The graphical representation in Figure-1 shows high slope at the extreme ends that is from case-1 to case-3 and from case-19 to case-21 results with outstanding and below average cases. Most of the cases lie in good and average categories. And Figure-2 shows most of the points between 1.5 and 2.5 results for good and average cases.

Using proposed metrics and effort calculation it observed that more than 50% cases lies in good and average category. The comparisons of 5-Components with 10-Components illustrates that the above percentage increases from small to big projects.

6. CONCLUSION

CCM and CRM metrics proves itself as early stage metrics for the component based software development with knot model of component based software life cycle. These Metrics helps in determining the composition of the system and reducing efforts required for development of the whole software system. Further research can be carried out on these metrics by implementing other tools and methods instead of Likert 3-point scale rating for better results. There is large scope of research of calculating Effortin terms of time, cost, quality, operability, changeability, adaptability and other factors of software from time to time. More early stage metrics can be designed for good results and minimum efforts of software development using other life cycle models also.

7. REFERENCES

[1] Component software glossary, www.objs.com/survey/ComponentwareGlossary.htm

[2] Gary T. Leavens, Murali Sitaram, Foundations of component-based systems,Software Engineering, A practitioner's approach, Roger S. Pressman 5th ed.

[3] Boehm B., "A Spiral Model of Software Development and Enhancement", ACM SIGSOFT Software Engineering Notes, 1986, Vol. 11, No. 4, pp. 14-24.

[4] Gill N. S. and Tomar P., "X Model: A New Component-Based Model", MR International Journal of Engineering and Technology, 2008, Vol. 1, No. 1 & 2, pp. 1-9.

[5] Luiz Fernando Capretz, "Y: A new Component-Based Software Life Cycle Model ", Journals of Computer Science1 (1):pp.76-82.

[6] Srivastava, Chauhan, Raghuraj, "Square Model- A proposed Software Process Model for BPO base software applications",International Journal of Computer Applications, Vol 13, No. 7,2011,pp. 0975-8887.

[7] Chhillar, Kajla, "A New Knot model for Component Based Software Development", International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, May 2011 pg. 480-484.

[8] Barry Boehm. Software Engineering Economics. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[9] Barry Boehm, Chris Abts, A. Winsor Brown, SunitaChulani, Bradford K. Clark, Ellis Horowitz, Ray Madachy, Donald J. Reifer, and Bert Steece. Software Cost Estimation with COCOMO II . Englewood Cliffs, NJ: Prentice-Hall, 2000.