A Metrics Methodology for Predicting Reusable Suite Object in Component Based Software System

Kumar Rahul[1]
Assistant Professor, NIFTEM
Haryana, INDIA

Brijesh Kumar Sinha[2]
Assistant Professor, NITRA Technical campus
Ghaziabad, UP

Abstract

Component based software system is a new paradigm in software development process. Component based software system approaches to the rapidly development, deployment and evolution of different components of complex software system with minimum engineering effort and resource cost. Component based software system has to this objective is the reuse of software component suite in developing process of multiple system. To achieve this goal, software engineers must apply effective metrics method with modern tool within the context of a mature software process to evaluate reusable suite of software system. In addition that to predict high efficiency. CBSS demand software development to start from scratch with the help of metrics method which affect the efficiency, maintainability, reusability, and productivity and delivery time of the software. Our paper also explores the acceptance of reusability characteristics and sub-characteristics defined in ISO9126 quality model for CBSS.

Keywords: component, reusability, metrics, deployment, ISO9126.

I. INTRODUCTION

Component based software system is concerned with the new metrics methods for improving software reusability suite quality and enhancing development productivity for different system. This paper of component based software system (CBSS) provides engineers with opportunities to streamline their software development process all through its phases from analysis to maintenance and from project planning to project completeness. Software development techniques followed are now getting refined with concept of metrics approach of component based software. These techniques are evolving to certain new phase.

II. REUSABLE COMPONENTS ISSUES

Reliability, accuracy and efficiency are still the biggest problem in software engineering. In Components based software development, reliability of the software considered first during development.

Out of available components which one is effective to reuse during development time of software is an important activity for software engineer. Software engineer should choose the component, based on the characteristics defined in the terms of complexity, effectiveness for the software being developed. Transforming synchronous or asynchronous predicting reuse concepts with metrics method that will convert into a technology of implementation. It would provide the basis for the future scope of software domain for ensuring reliability, accuracy and efficiency for improving quality and increasing productivity for the software being developed. Currently all reuse occurs in the project development, where reuse is difficult because a project's focus is system delivery[3]. Our metrics will calculate the efficiency in terms of percentage to show effectiveness on every aspect of components being reuse. Metrics method for evaluating the reuse component process model (figure-1) is work on the request comes from system level and it proceed to component based level in software agency. When the requests (in form of functions, will appear which component should use) generate at system organization level then different view of measurability like requirement and analysis, specific-design, component identification and component integration, testing, quality control factor and finally release of component will estimate whether the components would predict the best use of component based software engineering principles[4]. The system organization performs activities specific to implementation of the project to which it is dedicated. It analyzes the requirement, specific-design that leads to high-level design of project to make the system organization reliable. However when the software developer as well as software engineer have identified the system component, usually after the so-called preliminary specific-design and testing process they request the component from system organization that will be responsible to maintain different categories of components and integrate them into the programs. Our metrics of different parameter judge the different components to integrate the components based on granularity, reliability, effectiveness and portability and the system they have designed[6]. The system organization software developer as well as software engineer may also request a list of components that satisfy a given specification. Then from the several design options they can choose the one for which metrics measurement will predict more useful in terms of understandability, adaptability and portability. The metrics measurement specifies the different parameter of software component in terms of reusability to make a project complete reliable and portability.

A. Regular activity

When the component levels receive a request from the system organization, its searches its categories of components with the help of metrics technique to find a software component that satisfies that request with or without tailoring[2]. Tailoring can be applied to a software component through two ways, either automatically and through modification. Modification requires a lot of changes in the request component and then submitting to component level. Modification works on approximate search of available components. When it finds the available reusable components through exact search of component at component level then it transfers the component to system organization[3]. The activity of development with reuse is realized by developing software with existing components. While reusing the components after metrics based prediction, reusability is the main characteristics to concern. Reusability is a parameter that measured in terms of degree to which a component can be reused, and reduces the software development process cost by enabling less writing. Component with higher reusability is a key issue for user to selecting the component among several reusable components. It is necessary to measure the reusable component suite through metrics methodology for estimating the components efficiency.

B. Abstracting components through a metrics component extraction model

As the numbers of metrics method approaches are available to predict the reusability developing component. It is generally more expensive and complex than developing specialized code, because of the overhead of maintaining the component level. Software metrics are intended to measure software quality characteristics quantitatively. Among several quality characteristics like understandability, adaptability, portability the reusability is particularly important when reusing components[5]. A rich and well-organized format of reusable components is the key to a successful component level and a long term economic gain from system organization. A fully complete application domain, where most of the functions that need to be used already exist in some form in earlier system, should provide enough components for code reuse[7]. To grouping such code for reuse, the component level analyzes existing programs in the two phases shown in figure 2. Components are stored in component repository system with all information that has been obtained about them. In the component identification phase, programs unit are automatically extracted, made independent, and measured according to observable properties related to their potential for reuse. It consists of four steps:

- Identification of metrics model. Using our current understanding of the characteristics of a potentially reusable component in our environment, we define a set of metrics model based on the different parameter to measure the predictability of component reuse in system organization level for implementation of the software. For different metrics it requires different acceptable range of values[1]. We verify the different metrics and their value ranges using the outcomes in the next steps and continually estimate them until we have an accurate estimation to count the predicting in terms of values that leads to select the specific component for reuse.

- Define metrics model. After identify the metrics model, based on our requirement, it would be available to reuse the component from component level to system organization. Whatever the number of request of component generated from system organization to component level it would be accepted first then evaluated for calculating the prediction in terms of

	PSU	RSU	IDC	IIDC	OIDC
C1	0.33	0.58	0.41	0.5	0.33
C2	0.66	0.50	0.66	0.5	0.75
C4	1	0.75	0.88	0.80	1
C5	0.50	0.66	0.50	0.50	0.50

comparable value to list out in reusability storage. Before we define metrics, we need to know the type of information that is available. Different metrics will have different set of values after calculation to show the predict ability of component.

- Extraction of components. We extract modular component from existing system. It will help, based on the characteristics of component. Modular component means a syntactic unit such as a C/C++ function, a block or subprogram or a FORTRAN subroutine.

- Application of the metrics model. For determining the values of reusability level implementation require the application of metrics model. Metrics based approach will come out in figure as result at application of metrics. The current reusability model is applied to the extracted, completed components. Components whose measurements are within the model's range of acceptable values become reusable components to be analyzed by the domain experts in the eligibility phase.

Domain expert analyze request reusable component to understand and record each component's meaning in different terms for reusability during the component eligibility phase. It also evaluating its potential for reuses in future system. Domain expert is basically a metrics system that evaluate at the level of the different parameter as functionality (i.e. suitability, accuracy, interoperability, security and compliance), reliability and usability.

Earlier model like component service utilization metrics have discussed in details but fails to show the predictability to make a component reuse in effective way.

III. EXISTING WORK

According to "metrics-based selection of a component assembly" proposed by CAMELIA SERBAN[(1)] AND ANDREEA VESCAN[(1)] , we found different result as:

TABLE 1: Data collected from above mentioned paper Where,

PSU-provided service component
RSU-required service component
IDC-interaction density of component
IIDC- incoming interaction density of component
OIDC- outgoing interaction density of component

EXAMPLE 1:- you have been appointed a project manager for a major software products company. Your job is to manage the development of a software as well as maintenance of the software. While working on different module of the software you have to reuse component from the existing software for developing as well as maintenance of the software. Suppose a system organization is having number of user input is 24, user output is 60 and 50 components of existing software are available. Each component is having different size. Out of which 12 is participating in between sharing of components between system organization and component level. Out of 12, only 5 components are having actual direct function to interact from system organization to component level. So actual interaction is 5 and maximum available interaction between them is 12. Actual incoming interaction from system organization to component level is 3. But maximum available incoming interaction among them cannot exceed to 6. The rest 6 component is maximum available outgoing components from component level to system organization. Out of 12, 2 is actual outgoing interaction from component level to system organization. 4 components are provided to component levels that are actually used by other module. 7 components are required for software2 that are actually using and total 9 components are required by software2. 1) Calculate the utilization of component in both the cases provided and required. 2) Calculate complexity or density in terms of incoming as well as outgoing of components.3) find SICCM and AICSCM with the help of different components.
Solution:

1) $P_{actual=}$ 4 and $P_{total=}$12

So, PSU= P_{actual} / P_{total}
= 4/12
= 0.33

2) $R_{actual=}$ 7 and $R_{total=}$ 12
So, RSU= R_{actual} / R_{total}
= 7/12
=0.58

3) I= 5 and $I_{max=}$ 12
So, IDC= 5/12
=0.41

4) $I_{in=}$ 3 and $I_{maxIN=}$6
So, IIDC= 3/6
=0.5

5) I $_{out=}$ 2 and $I_{maxOUT=}$ 6
OIDC= 2/6
=0.33

Now, instead of these we have proposed different metrics (based on reusable component process model and component extraction through metrics) as follows:
PCUM- provided component utilization metrics
RCUM- required component utilization metrics
ICCM- interaction complexity of a component metrics
SICCM- sharing interaction complexity of component metrics
AICSCM- average interaction complexity of software component metrics

Now,

IV. DESIGN OF A METRICS METHODOLOGY TO EVALUATE THE PREDICTING OF REUSABILITY

A. Provided component utilization metrics:-

It's a basic metrics that has to work initially to count the number of component utilization. This kind of metrics will estimate the number of provided component utilization between two entities.

The PCUM represents the ratio of component provided by the system organizations which are actually used.

So, $PCUM_{Y=}$ $P_{REAL/}$ (P_{FINAL}-1).

Where $P_{REAL=}$number of component provided by the system organization y that are actually used by other system organization and $P_{FINAL=}$number of component provided by system organization y.

Here, $P_{REAL=}$ 3 and $P_{FINAL=}$ 5

So, $PCUM_{Y=}$ 3/4

=**0.75(higher in case of provided**)

B. RCUM (required component utilization metrics) is similar but for required services.

$RCUM_{Y}$= $R_{REAL/}$ (R_{FINAL}-1)

Here, $R_{REAL=}$ number of component required by the system organization y that are actually provided by the assembly and $R_{FINAL=}$ number of component required by system organization y.

So, $R_{REAL=}$ 7 and $R_{FINAL=}$ 9

So, $RCUM_{Y=}$ 7/8

=**0.87(higher in case of required)**

C. Interaction complexity of component metrics (ICCM) – It define as a ratio of actual interactions between system organization and component level. i.e.

ICCM=# AI/ # (AI $_{max}$-1)

Where # AI= actual interaction between system organization and component level

And # $AI_{max=}$ maximum available interaction of component between system organization and component level.

Here, AI=5 and $AI_{max=}$ 12-1=11

So, ICCM= 5/11

= **0.45(Its higher than IDC value)**

D. Sharing interaction complexity of component (SICCM):-

This metrics will estimate the complexity of component interaction between system organization and component level. When the request generated at system organization, it will evaluate at component level with the help of number of component is exchanging.

So, the SICC= # I_{IC}/# I_{maxIC}

Where # $I_{IC=}$ actual sharing interactions and

I_{maxIC} = maximum available sharing interactions

Where # $I_{IC=}$ 5 and # $I_{maxIC=}$6

So, SICCM= 5/6

=0.83(this value is higher than both the values of incoming interaction density of component (IIDC) and outgoing interaction density of component (OIDC)). We are not giving any specific incoming and outgoing density of components; instead of these we are giving a common interaction complexity of component that is SICCM, where the result will be higher from both the case.

E. AICSCM-Average interaction complexity of software component metrics. (It's an additional approach. It was not there in earlier case.)

AICSCM- ICC1+ ICC2+--------------+$ICC_{N/\#}$ COMPONENTS available

AICSCM for each component divided by the total number of component

So, ICC1-0.45(from above)

#component= 12, so AICSCM= 0.45/12

= **0.0375**

Now, new result of component CA can be shown in below Table2, with the result of other component of Example 2.

V. EXAMPLE AND RESULT ANALYSIS

EXAMPLE 2:

Referring to figure 1, based on the results of problem 1, you have been appointed a project manager within an information systems organization. your job is to build an application that is quite similar to others software system your team has build, although this one is larger and more complex, but utilization factor of different specific component such as component1, component2,component3 and component4 and so on will have highest probability metrics than the other existing software. A different set of values for different components has given in below table. Now

1) Calculate the utilization of component in both the cases, provided and required for C_B, C_C AND $C_{D.}$
2) Calculate interaction complexity of each component such as ICCM.
3) Find SICCM and AICSCM with the help of different components.

	C_B	C_C	C_D
P_{REAL}	3	3	4
P_{FINAL}	5	4	6
R_{REAL}	3	5	6
R_{FINAL}	4	7	8
# AI	4	4	3
# (AI $_{max}$-1)	5	4	4
# I_{IC}	5	6	4
# I_{maxIC}	6	7	6

TABLE 2:Different values of participating component between system organization and component level through different aspects.

Where,

$P_{REAL=}$number of component provided by the system organization, that are actually used by other system organization.
$P_{FINAL=}$number of component provided by system organization y.
$R_{REAL=}$ number of component required by the system organization y that are actually provided by the assembly.
$R_{FINAL=}$ number of component required by system organization y.
#AI= actual interaction between system organization and component level.
#$AI_{max=}$ maximum available interaction of component between system organization and component level.

$I_{IC=}$ actual sharing interactions and
I_{maxIC} = maximum available sharing interactions.

Comparisons of metrics result after applying the different strategies of develop metrics. It is evaluated

in terms of threshold values and its limit between values 0 and 1.
We got these values as,

	PCUM	RCUM	ICCM	SICCM	AICSCM
CA	0.75	0.87	0.45	0.83	0.037
CB	0.75	1	0.80	0.83	0.055
CC	1	0.83	1	0.85	0.208
CD	0.80	0.85	0.75	0.67	0.110

TABLE 3: Final comparable result from value from table 1

VI. CONCLUSION AND FUTURE WORK

In this paper, we have presented different metrics methodology approach to predict the reusability concept for better improvement from the existing approach given in “metrics-based selection of a component assembly” by CAMELIA SERBAN(1) and ANDREEA VESCAN(1). In order to assess , in an quantitative manner, some quality attributes (in terms of component sharing between different system’s architecture) that are considered important for our system . The PCUM (provided component utilization metrics) and RCUM (required component utilization metrics) approach is the necessity metrics for any component at initial stage of finding out predictability, so that we can stimulates the other parameter regarding reusability accessibility. we have proposed a customized approach that work to predict where a number of components are working together So we are having a metrics methodology that is predicting the higher reusability in component based software engineering. We have planned a architecture that will work in sharing interaction complexity of component (SICCM) there result will evaluate in terms of actual sharing interactions and maximum available sharing interactions of different component. At the end of metrics we have given an average concept of proposed metrics methodology, AICSCM (average interaction complexity of software component metrics), this metrics approach evaluate the combined reusability parameter predictability for developing software. Future work in this respect is controlling and monitoring of the different metrics methodology for prediction of reusable as well as different parameter as functionality, reliability, understandability.

REFERENCES

[1] V. L. Narasimhan and B. Hendradjaya “A New Suite of Metrics for the Integration of Software Components,”

[2] Mahmood, S and R. Lai. RE-UML. An extension to UML specifying Component-Based Software System, Software Engineering Conference,220-228 (2009).

[3]Capra E., Francalanci C., Merlo F. (2008), “An Empirical Study on the Relationship among Software Design Quality,Development Effort, and Governance in Open Source Projects”, IEEE Transactions on Software Engineering, Vol. 34, No. 6, Nov/Dec 2008, pp765-782.

[4]Emanuel A.W.R, Wardoyo R., Istiyanto J.E., Mustofa K. (2010), “Success Rules of OSS Projects Using Datamining 3-Itemset Association Rule”, International Journal of Computer Science Issue (IJCSI), Vol.7, Issue6 Nov.2010, pp71–80.

[5]P.S. Sandhu, H. Singh, “Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach”, International Journal of Information Technology, vol. 3, no. 3, pp. 209-214, 2006.

[6]G. Wang, R. Valerdi, J. Fortune, “Reuse in Systems Engineering”, IEEE Systems Journal, vol. 4, no.3,pp376-384,September2010.

[7]CH.V.M.K.Hari, P.V.G.D.P Reddy, J.N.V.R.S Kumar, G.SriRamGanesh. CH.V.M.K. Hari., “Identifying the Importance of Software Reuse in COCOMO81, COCOMOII”, International Journal of Computer Science and Engineering JCSE, vol. 1 no. 3,pp142-147,2009.