PROCESS MODEL FOR REUSABILITY IN CONTEXT-SPECIFIC REUSABLE SOFTWARE COMPONENTS

V. Subedha*

Research Scholar, Department of CSE, Sathyabama University
Chennai, India
subedha@gmail.com

Dr. S. Sridhar

Research Supervisor, Department of CSE, Sathyabama University
Chennai, India
drssridhar@yahoo.com

Abstract

Constructing component based software using reusable components is becoming a promising approach. Context-specific reuse is a broadly used way to increase the value of reuse. This paper reports our on-going work aimed at reusing context-specific reusable software components from an existing system to achieve large scale reuse. Since the quality of a context specific reuse system is highly dependent on the quality of the component, quality analysis becomes very critical to the reuser. The proposed process model adopts the quality hierarchy process method to determine the reusability factor and makes the quality assessment of the reusable component more effectively for better reusability. We discuss a dynamic approach to analyze the component for reusability. The basic premise assumed by this model is that reusable components have certain quality factors like functional coverage report, time, functional usefulness etc., and which can be measured to certain extent with help of metrics. This process model combines three quality factors into reusability metrics based on the quality hierarchy process. Here, we also introduce a new algorithm for the process of reusing the context-specific components from the existing system

Keywords: Reusable Software Components, Reusability, Reuse Metrics, Extraction Time, Component Identification, Component Qualification, Reuse libraries.

1. Introduction

A Reusing software component is becoming a widely used approach to reduce the software development costs and shorten the software development production cycle. Software reuse can be applied to any life cycle product. Graaf et.al [1] identifies ten potentially reusable assets of software products as shown in the Fig 1.

Context specific system may compose of hundreds of reuse components and if the quality of just a one component is poor, it may impact the overall quality of the entire system. Any defective component which is reused could cause a ripple impact in the entire existing system. Therefore, effective analysis for component quality needs an adequate process model and factors for reusability.

The typical objectives of Process model for Reuse are to:

- Quality analysis for reusability based on the factors
- Enable reuser to reuse quality components stored in Component repository.

This paper proposes a process model for reusability in context specific level based on the quality factors like functional usefulness, time and reuse frequency of the components. Moreover, this paper proposes a dynamic and algorithmic approach to analyze the component for reusability. In addition, it presents an application example to demonstrate how to use the proposed dynamic approach to support reusability process.

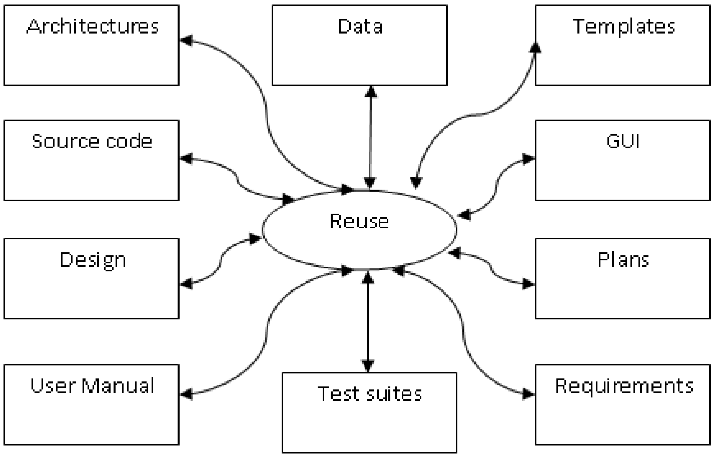

Fig. 1. Reusable assets of software products

The fundamental motivation of the proposed approach is to reduce the amount of data that the reuser has to review in order to find out context specific reusable software components. The reuser has to analyze only a few numbers of proposed candidates which, if accepted for reuse become then the foundation of reusable components. This process model is able to detect the "Excellent" components for reusability. This model is able to detect 30% to 40% of excellent components from the set of good components.

The remainder of this paper is structured as follows. Section 2 illustrates some related works in the existing literature. Section 3 presents the proposed process model for context specific reusability. Section 4 & 5 deals with the factors and metrics involved in measuring the reusability. In Section 6, we conclude this paper.

2. Related Work

Although there are many numbers of published papers addressing the software reusability, only few of them address model for reusability. In this section we review some models based on the software reuse metrics and discuss how they are related to our proposed model. The contributions of metrics to the overall objective of the software reuse is very well understood and recognized. But how these collectively determine reusability of a software component is still at research stage. The most common proposed approach was to define metrics to assess the reusability.

Cho et al [2] proposes a set of metrics for measuring various aspects of software components like complexity, customizability and reusability. The work considers two approached to measure the reusability of the component. In first approach the reusability is measured in the design phase in a component development process and in second approach particular component's reuse level per application in a component based software development is measured.

Washizaki et al [3] discusses the importance of reusability of components in order to realize the reuse components efficiently and propose a component reusability model from the view point of component reuser.

Richard W. Selby [4] identified factors that characterize module reuse are low coupling, high cohesion and discussed that CK metric suit is able to target all the essential attributes of OO-based software. Husein and Oxley [5] describe coupling and cohesion metrics suite is presented to evaluate object-oriented software. These metrics may be applied to assess reusability.

Parvinder Singh and Sandhu and Hardeep Singh, [6] [7] have used metric based approach for identifying a software module and the reusability was obtained with the help of Fuzzy Logic and Neuro-Fuzzy. This research shows how metrics can be used to identify the quality of a software component.

Himani Goel and Gurbhej Singh [8] proposed an Expectation Maximization based clustering approach to evaluate the reusability prediction of function based software systems. Here, the metric based approach is used for prediction. The metrics used for measuring the reusability of software modules are Cyclometric complexity, volume, regularity, reuse frequency and coupling.

Sonia Manhas et al [9] proposed Reusability Evaluation model for procedure based software systems to calculate reusability value which enables to identify a good quality component for reuse. The framework of metrics proposed for this model are Cyclometric complexity, volume, regularity, reuse frequency and coupling. Neural Network techniques are explored to design the reusability evaluation model.

Parvinder & Shalini [10] proposed particle swarm optimization technique along with the four variants of conjugate gradient algorithm to train the feed forward network. The performance of the train neural networks is tested to evaluate the reusability level of the procedure based software systems.

Ajay Kumar [11] propose Classification of the reusability of software components using Support Vector Machine and also the identification of reusable software modules in Procedure Oriented System is based on software metrics like Cyclometric complexity, Volume, Regularity, Coupling and Reuse frequency.

In all existing model the static metrics are defined to evaluate the quality of the components for reusability where we propose dynamic metrics to evaluate the quality of the components for reusability.

3. Process Model for context-specific reusability

In this section we introduce an algorithmic approach for reusability for context-specific components from the existing system. Fig. 2 depicts the 15 steps of our approach in the pseudo-code format.

```
Step 1 : Begin

Step 2 : The reuser describes the goal of the reuse

Step 3 : The functional requirements for the reusability is collected

Step 4 : Generate the test cases , executes and collects the functional coverage report

Step 5 : Select metrics for identification and choose its bounds

Step 6 : Identifies the candidate components which satisfies the criteria defined in Step 4

Step 7: Update the dynamic component metrics library

Step 8 : Find out the optimal path for the identified set of components

Step 9 : Calculate the extraction time using METF method for each component in the identified set.

Step 10 : Calculate the reuse frequency for each component

Step 11 : Update the dynamic component metrics library with the attribute extraction time and reuse frequency.

Step 12 : Select metrics for qualification and choose its bounds

Step 13 :  Classify the indentified set into two parts as qualified and not qualified which satisfies the criteria defined in Step 12

Step 14 : The resuer makes the final decision about the reuse of the components in the qualified set.

Step 15 : End
```

Fig. 2. Pseudo-code format for reusability process model for context specific components

We begin the process for reusability for context-specific in step 1. In step 2, the reuser formulates the goal and describes functionality of the components which they want to reuse from the existing environment. In the next step functional requirement for the reusability is collected. In step 4, the reuser generates and executes the testcases and functional coverage analysis report was collected. In next step reuser select metrics to measure the functional coverage analysis and choose the lower and upper bounds. In step 6, we identify a set of components for reusability based on the metric criteria developed in step 5. The next step is to update the dynamic component metrics library which is dynamically created for each reuser's requirements.

In step 8 & 9, the process is to find out optimal path for extracting the components and also to calculate the extraction time based on the method Minimum Extraction time First (METF). In next step the main focus of the reuser is to decide the functional usefulness of the identified components. So calculate the reuse frequency using the formula in Eq. (1). In step 11, update the dynamic component metric library with the extraction time and reuse frequency which is calculated in step 9 and 10.

In the next step, we select all the metrics which defined in table 1 and choose it's bound to measure the quality of the component to be reused. In order to achieve the goal, by applying the filtering strategy based on the bounds in step 12, we classify the identified set into two parts as qualified and not qualified. This process is called as qualification and it is done in step 13. In step 14, using the qualified set the reuser decided whether to reuse the components or not. Their decision is influenced as well by factors such as performance.

4. Factors involved in measuring the reusability

The quality hierarchy process is a theory of measurement through the factors for reusability. Structure the problem as a hierarchy in the form of Fig.3. The hierarchy begins at the top most level with an overall objective to be achieved. The next level under the objective is to select the factors involved in measuring the quality of the components. The lowest level consists of the various metrics to meet the objective

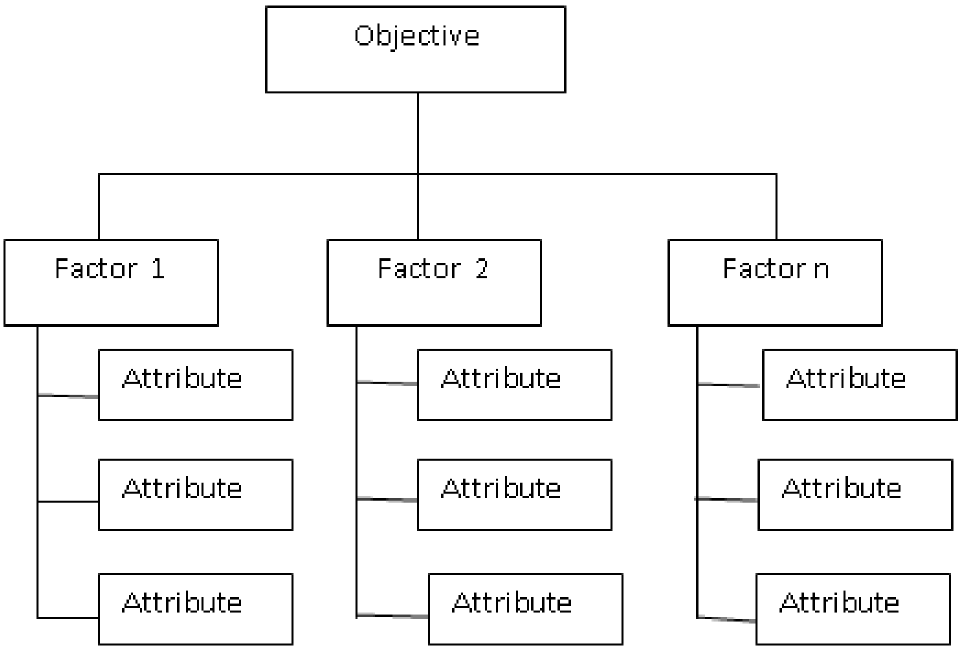

Fig. 3. Generic Quality hierarchy Structure

The following is process to determine how much each factor would contribute to reusability in context specific. Structure the quality hierarchy in Fig 4. The objective is to determine the high quality and potential benefit components for reusability. The next level under the objective is the set of factors which contribute to reusability. The lowest level consists of various metrics that meet each factor. It can be observed from the Fig. 4 that reusability depends on functional coverage analysis report, time and functional usefulness. Each of these factors is explained below.

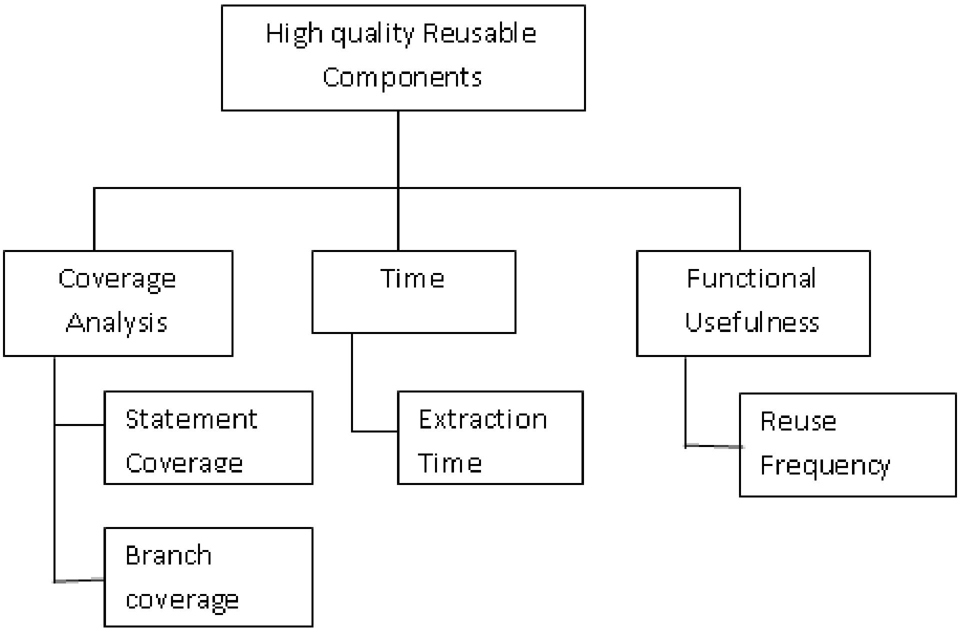

Fig. 4. Reusability hierarchy structure for context-specific

Coverage analysis : Coverage is used as a goal, by which the testbench and test suite will be measured throughout this verification process. The approach saves significant human and machine resources, shortens time-to-market and eventually contributes to a mature and a high-quality product. Performing coverage analysis is to decide whether the component has good quality or not.

Time : Time includes the time needed to extract the component from the identified set in the existing environment. Measure of time provides a partial qualifier in order to achieve the reuse in terms of speed. That is, the components which are nearby to the resuer are usually easier to extract in terms of speed.

Functional usefulness : To be reused, a precondition is that the functionality of the component must be useful for the reuser. It is extremely hard to decide the usefulness of the component since it is based on the domain knowledge and the requirement of the reuser. However, an indirect automatic measure of usefulness was development to measure the reusability of the component. The highly used components within in an environment are an excellent candidate for reuse in context level. We exclude the components that are not frequently used in the existing environment.

5. Metrics for Coverage analysis, Time and Functional usefulness

Our motivation of this section is identify suitable metrics with lower and upper bounds to support the classification of the components as excellent or good candidate for reusability. Four primitive metrics are selected to measure the three factors. Table 1 contains the definitions of the metrics used for measuring the coverage analysis, time and functional usefulness.

Table 1. Definitions of Metrics to measures the factors

Metrics	***Definition***
Statement coverage	This metric reports whether each executable statement is encountered
Branch Coverage	This metric reports whether Boolean expressions tested in control structures
Extraction Time	This metric is used to measure the Extraction time of each component
Reuse Frequency	This metric is an indirect measure of the functional usefulness of a component

Measuring coverage analysis report : Using the functional specification the reuser generates a set of test cases and the test cases are executed. The functional coverage report of the component is collected. Three commonly used measures of coverage driven functional verification are statement coverage, branch coverage and logical path coverage. We use the statement coverage and branch coverage as metrics for coverage analysis to qualify the candidate component for reusability.

Measuring time : The components having the extraction time less than the average extraction time is qualified for reuse. The reason for choosing the extraction time as metrics is to speed up the process of reuse. The extraction time and the optimal path for the extraction are calculated using a scheme called as minimum Extraction Time First.

Measuring functional Usefulness : The reuse frequency is an indirect measure of the functional usefulness of a component. We measure the functional usefulness that frequently used system is a good candidate for reuse in context level in similar domain. Hence we choose the metrics reuse frequency as a qualifier for classifying the components. Reuse frequency of each component can be calculated using the equation (1).

$$\text{Reuse Frequency} = \frac{n(C)}{\frac{1}{n}\sum_{i=1}^{n} n(Si)} \quad (1)$$

where n(C) is total number of reference to the Component, , n(Si) is total number of reference for each Standard Components in the existing environment & n is the total number of component in the existing environment.

6. Conclusion

The fundamental motivation of this process model is to reduce the effort spent by the resuer to identify and qualify the component candidates for quality reusability. The basic factors used for indentifying and qualifying the component for reusability is like coverage analysis report, extraction time and functional usefulness that can be broken down into an measurable items. This paper gives an insight view of reusability metrics for component based system and empirical evaluation of these metrics on various generic components. The next step is to apply the proposed process model on large-scale systems to analyze the benefits and limitations. Also it is hoped that in future, we will be able to expand the list of reusable factors and some new enhanced metrics which will provide more complete measures.

References

[1] B. Graaf, M. Lormans and H. Toetend, “Embedded Software Engineering: The state of the Practice “, IEEE Software, Vol. 20, No. 6, pp.61-69, Nov 2003.

[2] Eun Sook Cho, Min Sun Kim, Soo Dong Kim, “Components Metrics to Measure Component Quality”, Proceedings of the Eighth Asia-Pacific Software Engineering, pp. 419 - 426, 2001

[3] Hironori Washizaki, Hirokazu Yamamoto and Yoshiaki Fukazawa, “Software Component Metrics and its Experimental Evaluation”, Proceedings of the International Symposium on Empirical Software Engineering (ISESE 2002), Oct 2002.

[4] Richard W. Selby, “Enabling Reuse-Based Software Development of Large-Scale Systems,” IEEE Transaction of Software Engineering, Vol. 31, No. 6, PP. 495-510, Jun 2005

[5] Husein. S and A. Oxley, “A Coupling and Cohesion Metrics suite for Object-Oriented Software”, Proceedings of International Conference on Computer Technology and Development, Malaysia, pp.421-425, Nov 2009

[6] Parvinder Singh Sandhu and Hardeep Singh, “A Fuzzy-Inference System Based Approach for the Prediction of Quality of Reusable Software Components”, International Conference on Advanced Computing and Communications, ADCOM 2008. on page(s): 349 – 352

[7] Parvinder Singh Sandhu and Hardeep Singh, “Automatic Reusability Appraisal of Software Components using Neuro-Fuzzy Approach”, International Journal Of Information Technology, vol. 3, no. 3, 2006, pp. 209-214.

[8] Himani Goel and Gurbhej Singh, “Evaluation of Expectation Maximization based Clustering Approach for reusability Prediction of Function based Software Systems”, International Journal of Computer Applications, Vol 8,No. 13,pp.13-20,Oct 2010.

[9] Sonia Manhas, Rajeev Vashisht, Parvinder S. Sandhu and Nirvair Neeru,”Reusability Evaluation model for Procedure Based Software Systems”, International Journal of Computer and Electrical Engineering”,vol.2.,No.6, pp.1107-1111,Dec 2010.

[10] Parvinder S. Sandhu and Shalini Chhabra,”A Comparative Analysis of Conjugate Gradient Algorithms & PSO Based Neural Network Approaches for Reusability Evaluation of procedure Based software Systems”, Chiang Mai j. Sci, Vol 38(special issue), pp.123-135, Jan 2011.

[11] Ajay Kumar, “Measuring Software Reusability using SVM based Classifier Approach”, International Journal of Information Technology and Knowledge Management.”, Vol. 5, No. 1, pp.205-209, Jan 2012.