Classification and Selection of Reusable Software Components Using Mutation Technique

[1]R. Kamalraj, [2]Dr. A. Rajiv Kannan, [3]P. Ranjani
[1]Dept. of CSE, SNS College of Technology, Coimbatore, Tamil Nadu, India
[2]Dept. of CSE, KSR College of Engineering, Tiruchengode, Tamil Nadu, India
[3]Dept. of IT, Avinashilingam University for Women, Coimbatore, Tamil Nadu, India

Abstract
In Software Engineering classifying reusable components for further use is an important activity to reduce the required efforts for constructing the total system. Existing methods are available to categorize the components as per their name and applications only. So in this paper we focused on 'Mutation Technique' from Genetic Algorithm approach to categorize the reusable software components as per their features. By using 'Mutation Technique' system development may get a different 'Reusable Component Recording System' to select the components to improve the quality of the system.

Keywords
Reusable Components; Genetic Algorithm; Package Classification; Mutation

I. Introduction

In System Engineering, construction of a new system is based on many software engineering activities such as 'Requirements Engineering', 'Designing', 'Developing code', 'Testing code' and 'Maintenance Engineering'. In the first step, the 'Project Development Cost' should be estimated for assigning milestones in the product development. This activity will require existing product and reusable components details to reduce the development cost as much as possible [1]. When reusing the components from the existing products the different abilities of the component(s) should be analyzed deeply [8]. In this paper we propose the concept using Genetic Algorithm approach to categorize the reusable packages or components for easy use in upcoming product developments [7].

A. A View on GA

Genetic Algorithm Approach can be applied in 'Package Reusability Engineering' to classify the components for further use in product development [5]. The mutation technique can be applied to identify the feature of the components for matching with the user requirements in the product. To analyze the packages chromosomes pattern may be required to find the quality of the reusable component.
The 3 main steps are in GA

- Assumption - Consider the existing package features are base chromosomes
- Identifying Partner Chromosomes - different features to be added with the existing component.
- Combining - Combining base and Partner chromosomes to form a new component.

1. Available Software Features

The base chromosomes may be taken if any of their features can satisfy the user requirement.
That can be represent like,
Base Chromosome = (Requirement == Available Features)

2. Partner Chromosomes – Improving

In this step, finding the new partner chromosomes will have the valuating the existing features with the current product development requirements. If new requirements cannot be satisfied with the available package features then new requirement considered as a 'Partner chromosome' to improve the level of the RC.

3. Combining Task

After Identifying the Partner Chromosomes, the base and new chromosomes (features) will be combined together to create a new chromosome (feature) that may give an updated service of the existing package elements.

II. Reusable Components

Software Component is a collection program elements to perform specified tasks for satisfying the execution of a software system. When already used component consider for new software product development then different features of the component must match the requirement [4]. This kind of component is known as 'Reusable Component'. To identify reusable components from existing products 'Domain Analysis' will be performed [10]. After that similar domain packages may be selected and applied in current product development. The Identified Reusable Component should not affect the quality of the system and system development.
Different quality factors of Reusable Component (RC) are

- Component Performance
- Required Memory Capacity to execute
- Features of Component (services level)
- Design Metrics of RC

A. Component Performance

Component performance states that how long the component can execute without fault. A component may be required to run for 24x7 level. So that RC must have a 'Stability' feature to support the system execution.

B. Memory Capacity

It is a very important factor to decide the speed of getting response from the RC. It is based on the number of program elements and the KLOC of Reusable Component.

C. Features of RC

In this, the functionalities of RC will be analyzed to prefer in the upcoming product engineering.
If it can full fill some extent of requirements then only it may be prefer to acquire for further product construction.

D. Design Metrics of RC

Design Metrics of Reusable Components are 'Coupling', 'Stability', 'Cohesion' and 'Complexity'. Values of each metric will be analyzed clearly to rank the package level for reuse [2].

III. Mutation Technique

Mutation Technique from Genetic Algorithm can help to keep the information of packages in 'Reusable Component Recording System'. Mutation states that there is a 'Alteration on chromosome patterns' or 'Changing Number of Chromosomes' in DNA structure. This method can be applied on reusable components to organize the reusable components effectively.

A. Alteration on Chromosome Patterns

This method may be preferred to replace the faulty chromosome pattern with advanced or new chromosome pattern to keep the original chromosome pattern. In RC if any of the part is not working properly then that affected program element may be replaced with created new program element to provide the specified functionality in SRS(Software Requirement Specification) document. Simply it is 'Replacement Technique' like 'Corrective Maintenance' on RC.

This can be stated like this.

Algorithm 1:

Step 1: Finding 'Faulty Chromosome or Feature' of available RC.

Step 2: Developing the same feature with a new program element.

Step 3: Replacing the identified 'Faulty feature' program element with created new one.

The performance of the same RC may be improved by performing this method. Because, the new program element may not have any errors and it may have very less program instruction to carry out the task assigned on the RC.

B. Changing Number of Chromosomes

In this method, the number of features of the available RC may be improved or reduced as per the current product requirement. Here we have two cases.

Case 1: if only specified features are required from the RC then only that particular features' program instruction needed to exist in the RC.

This 'Case 1' is focusing on reducing or having same number of features to keep in the RC. From applying this activity only proposed features program element may be required to exist in the component. So size of 'Reusable Component' is changed dynamically in product engineering. It should not affect the performance of the system and improve the performance of the system.

Case 2: if new features are going to add with the existing RC then retain the existing features with new features

The second case is stating that the existing identified RC may be improved slightly by adding new features.

New feature may be in any of following form,

- Quality Factor
- Improved User-Interface Design
- New program element for new requirements

By applying this method, the number of chromosomes (features) in the component may be increased. When this 'case-2' is considered to be deployed then the Quality of the same RC should not be violated. So development team has this responsibility to secure the Quality of 'Reusable Component'.

IV. Reusable Component Categorization

To categorize the RC the Project Management Team has to maintain the 'List of Reusable Component (LRC)' for future project development. If a component is identified for in new product development at first time then it can be categorized as 'Normal RC'. So the level of that package has to be represented like 'Level 0'. The level-0 means that there is no modification in the identified 'RC'.

After some product development if the existing product(s) component(s) are considered to develop with little improvement in features then it will be represented as 'X-RC' of 'Normal RC'. This 'X-RC' will have the same program elements of 'Normal-RC' with additional program elements for new features. Here the new upgraded RC is named as 'X-RC' due to its power on satisfying product requirement.

Algorithm 2: To categorize the RC

Step 1: Scanning LRC to find RC

Step 2: If 'Normal RC' does not exist then
increment the no. of NRC by 1
record the details of NRC in RC Table format
else
increment the no. of X-RC of that corresponding RC NRC by 1.
increment the level of X-RC

Step 3: Procure the RC component as per the project management requirement.

If quality of the system must be improved
then
select X-RC of NRC
else
select NRC only

Here we suggested the table format for recording the details of Reusable Component of software systems.

Table 1: Example Format of LRC for the Products

Product Name	No. of RC	No. of X-RC
Product1	3	1
Product2	2	1
Product3	3	2

Table 2: Example Format of LRC for the RC in the Product

S.No.	RC Name	Level Indicator	X-RC Name	Level of X-RC
1	Login	0	Biometrics based Login	1

Here we suggested that format of keeping data about RC for identification and prefer for further usage in the project development. Package Table or RC Table will keep additional details for performance requirement. Assigning 'X-RC' name to the new RC must have good quality in all specifications. It should not disturb the quality of the entire system.

V. Results and Discussions

In software engineering field there are different methods are available to keep product information with reusable component information [6,9]. But this paper is focusing on 'Mutation' technique to differentiate the upgraded reusable component with the available Normal Reusable Component. So providing good quality in system functions with identified RC this approach can help in a good manner. The Reusable Component name is prefixed with X to categorize it as 'RC with Advanced Features'. Separating 'Base' and 'Advanced' Reusable Component may assist with project development team to reduce the development

cost, improve the productivity and enhance the quality of total system [3]. As well as controlling number of X-RCs entries in LRC of Base Normal Component is required to reduce the load on scanning LRC.

VI. Conclusion

The proposed method of classifying reusable component can give good details of RC for easy access. Applying the biology based classification and procurement of RC may reduce problems in Resource Management and Change Management activities. From identifying 'X-RC' in existing system can improve quality of work and improve the quality standard of the new product. But it requires additional time to record the details and memory space for keeping reusable component of same features.

References

[1] Wang A.J.A,"Reuse Metrics and Assessment in Component-Based Development", Proceedings of Software Engineering and Applications, Vol. 47, pp. 693-707, 2002.

[2] R.Kamalraj; A. Rajiv Kannan; R. Hemarani,"Metrics of Reusable Packages for Enhancing the Blocks of Software Project Management", CIIT International journal of Software Engineering and Technology, August, 2009.

[3] Nunamaker J.F.; Chen M,"Software productivity: a framework of study and An approach to reusable components", System Sciences Software Track, 22nd Annual Hawaii International Conference Proceedings, Vol. 2, 3-6, 1989.

[4] Nancy Bazilchuk; Parastoo Mohagheghi,"The Advantages of Reused Software Components", R&D and Technology Transfer, 2005.

[5] Byung-Jeong Lee; Byung-Ro Moon; Chi-Su Wu ,"Optimization of multi-way clustering and retrieval using Genetic algorithms in reusable class library", Dec 2-4, Software Engineering Conference, 1998.

[6] Navneet Kaur; Jaspreet Singh Budwal,"Hybrid Approach to Retrieval of Reusable Component from a Repository Using Genetic Algorithms and Ant Colony", Proceeding of the International Conference on Genetic and Evolutionary Methods, Las Vegas, USA, 2008.

[7] Rajesh Bhatia; Mayank Dave; R. C. Joshi,"Retrieval of Most Relevant Reusable Component Using Genetic Algorithms ", Software Engineering Research and Practice -SERP, pp. 151- 155, 2006.

[8] Caldiera, G.; Basili, V.R,"Identifying and qualifying reusable software components", Feb, IEEE Computer Society, 1991.

[9] Parvinder Singh Sandhu; Janpreet Singh; Hardeep Singh,"Approaches for Categorization of Reusable Software Components", Journal of Computer Science pp. 266-273, 2007.

[10] Haeng - Kon Kim ,"Automatic Identification of Potential Reusable Mobile Components International Journal of Computer Science, Nov 2008.

R. Kamalraj received his B.E. degree in Computer Science & Engineering from Bharathiyar University, Coimbatore, Tamil Nadu, INDIA in 2002, the M.E. degree in Computer Science & Engineering from Anna University, Chennai, Tamil Nadu, INDIA in 2009, and pursuing Ph.D. degree in Software Testing and Quality Management at Anna University of Technology, Coimbatore. He is having 9 years of teaching experience in 4 different engineering colleges. At present he is working as an Assistant Professor in the Department of Computer Science and Engineering at SNS College of Technology, Coimbatore. His research interests include Software Testing, Software Quality Management and Data Mining.

Dr. A. Rajiv Kannan received his B.Tech degree in Information Technology from Periyar University, Salem, Tamil Nadu, INDIA in 2002, the M.E degree in Computer Science & Engineering from Anna University, Chennai, Tamil Nadu, INDIA in 2004 and the Ph.D in Computer Networks from Anna University, Chennai, Tamil Nadu, INDIA in 2011. He is having 7 years of teaching experience in engineering college at KSR College of Engineering and present he is a Professor & Head of the department of Computer Science & Engineering at KSR College of Engineering, Tiruchengode, Tamil Nadu, INDIA. His areas of interest are Computer Networks, Mobile Computing and Data Mining.