Stability-based Component Clustering for Designing Software Reuse Repository

R. Kamalraj
AP / CSE,
SNS College of Technology,
Coimbatore, Tamil Nadu,
INDIA.

Dr. A. Rajiv Kannan
Prof / CSE
KSR College of Engineering,
Tiruchengode, Tamil Nadu,
INDIA.

P. Ranjani
AP / IT
Avinashilingam University for Women
Coimbatore, Tamil Nadu,
INDIA.

ABSTRACT
Software Engineering needs tools and concepts support to produce the systems as much as possible with good quality. To improve the tasks and make confidence in system development the 'Reuse' concept will be applied. To reuse the existing patterns or program elements such as components those should be organized in a well designed manner to reduce the time and stress for finding suitable 'Reusable Element' to integrate in the new system development. Here the proposed concept 'Stability Based Clustering' method is focusing on 'Stability' metric of component(s) and 'Clustering Analysis' of Data Mining. Using this, only stable component(s) are organized as 'Reusable Component' and with exterior details in 'Reuse Repository'. Searching time of finding a Reusable Component may be reduced to some extent and reduce stress on People side.

Keywords: Stability, Clustering Analysis, Designing Reuse Repository, Component Categorization

1. INTRODUCTION
Software Reuse is an important term to improve the productivity in software engineering field. Reuse may be on 'Design Pattern', 'Program Elements' or Tools. To reuse an item it must be classified in an effective manner to reduce the time taken for searching in the domain or in the repository. Reusability is one of the qualities required in software development side to reduce the stress on developing future projects. Using methods or algorithms the Identified Reusable element(s) should be organized in 'Reuse Repository'. Here we propose 'Data Mining' technique that may help to maintain the reuse repository with quality reusable components. To categorize the 'Reusable Component' many methods are available such as 'Domain Based Classification', 'Service Based Classification' and 'UI Based Classification'.

1.1 Domain Based Classification
In 'Domain Based Classification' components are analyzed as per their problem domains [3]. For example, 'Employee Payroll System' domain may have sufficient number of components at a particular moment. In future another product may focus on the same domain. So the newly identified functionalities from the existing will be developed. From those new components there may be a possibility to have new reusable components. Then those new and old components can be categorized based on the domain name. Then only that domain based reused repository can provide in future products focusing on the same domain.

1.2 Service Based Classification
In the second method, reusable elements are grouped as per the services provided by them [5]. Like 'Login with user name and password' and 'Biometrics Based Login' those are providing the support for securing the system from unauthorized users.

1.3 UI Based Classification
In 'UI Based Classification', the GUI (Graphical User Interface) or forms are considered for grouping them to support in future products UI design.

Grouping the reusable components is essential for further usage in product development. The repository should have top features to find perfect suitable elements for suitable product development. Reusing the existing item may give the time to quality management team to improve the quality of the product.

2. REUSABLE SOFTWARE COMPONENTS
In Software Project Management side finding acquiring existing suitable items to develop the current ongoing system is very crucial activity. For doing that activity enough time is required to scan the existing products, techniques and tools used for building the system. While designing a software system the 'System Designer' can able to recommend some elements in the proposed design can play as a 'Reusable Component' in future. To identify such a kind of elements they need to focus on 'Design Metrics' of designed elements and perform component analysis or objects in the proposed project.

2.1 Reusable Component Design Metrics
The following factors may be needed to check the 'Reusability Level' of a software element in the proposed design of a system. If the component has possible values on design metrics then only it may be preferred for further tracing from the development team. They are

[1] Coupling
[2] Complexity
[3] Stability and
[4] Quality

2.1.1 Coupling

Coupling defines the integration between software elements to do a particular user need [1] & [10] . Here the coupling of design element represents the strength of connectivity between software elements . Coupling is in different types. Among then 'Highly-Coupled' and 'Loosely Coupled' types are playing important role to group the components as 'Reusable' one or not. If a particular element is not fully depend on other elements in the system then it is in 'Loosely-Coupled' type of connectivity. So it may be considered as a 'Reusable Item' in that specified system [4].

2.1.2 Complexity

'Structural Based Complexity' may be applied to identify the cross communication among components and interior communication among the component elements to do the particular system requirement. If a component has an 'Excessive Complexity' then it may not be grouped as a 'Reusable One'.

2.1.3 Stability of Reusable Component(s)

Stability states that the life time of the component to satisfy the system requirement through its services. If any of the changes happened in that reusable component immediately the stability factor of that component has to be measured for keeping the same component as 'Reusable' one.

2.1.4 Quality of Reusable Component(s)

Quality of a 'Reusable Component' is measured based on the different abilities such as 'Portability', 'Maintainability' of that corresponding component. A reusable element must work on any system and in any situation without any fault.

The above four metrics must be analyzed before selecting a software element for reusing in further product developments.

3. PROPOSED METHODOLOGY

Data Mining is a method for analyzing bulk of data to extract the knowledge from them. Applying Data Mining on Software Engineering to simplify the data handling results reduced efforts and cost in various aspects [8] & [9]. Data Mining is giving very effective approach like 'Clustering Analysis' to group the elements as per the required data item.

To categorize the elements the following data are needed.

1. Domain of the component
2. Quality of the component
3. Cost of the component

3.1 Clustering Analysis

In Clustering Analysis, the elements have to be analyzed as per the data recommended for grouping them. In existing 'Coupling-based' method the 'Coupling' value and type of that considered for having clustering of reusable elements . From that here we propose 'Stability' also to be considered to have a good quality repository with stable components. So 'Design Metrics' of that particular component should be analyzed with correspond metrics' values. If any of them crossed the threshold level of the metric then that component may not preferred for reuse in product development.

In the 'Coupling-Based' approach, the type of the coupling may be considered only to categorize the reusable elements . But in this proposed approach from the 'Import' and 'Export' coupling of a component will be used to calculate the stability value of that corresponding component. If it is not having perfect stability value like '1' then it would not be classified as a 'Reusable' element. So this system is very strict based condition to group the identified reusable components.

3.2 Steps of Clustering Analysis

The 'Clustering Analysis' approach can be used for classifying the elements as per anyone of their characteristic. So, proper metric or property should be recommended to make them in a particular group. To perform Reusable Component Categorization using 'Clustering Analysis' the following steps are recommended.

Step 1: Collect the metrics and measure them for analyzing components
Step 2: Trace the metrics one by one
Step 3: Compare metrics value with threshold value of them
Step 4: Check the Stability of the Component
Step 5: Using Complexity and Stability factors find the Quality of the component
Step 5: Stable component may be select for 'Reuse'.

To measure the 'Stability' of the component, the coupling values like 'Import Coupling' and 'Export value' of the components have to be measured and using the following formula the stability can be calculated.

Instability = Import Coupling / (Import Coupling + Export Coupling)

The result of the above equation may be in the range like 0 to 1.If the 'Instability' value is equal to '0' then that corresponding component has a good stability. Then its exterior data such as 'Application Domain' and 'Services of the Component' should be collected for further consideration to record in the 'Reuse Repository'.

3.3 Stability-based Clustering Algorithm

The proposed technique is represented as following algorithm.

```
Start:   Check the availability of Components
Step 2:  Measure the Coupling of component(s)
Step 3:  if Coupling value >20
            Then neglect that component
             Go to step 1 or end
         Else
            If Coupling Type == Export Coupling
            Then
              Record that Component with its application
              domain data
            Else
                  Find the cost for modification
                  If high cost
                        Then avoid that component
Step 4: Confirm that component updated with valid details.
```

Table 1. Instability-value Range and Component Type

Instability value Range	Component Category
0.1 to 0.4	Little-Modification-Required
0.5 to 1	More-Modification-Required

The proposed methodology may classify the component as 'Reusable' one when it has stability equal to 1. So the 'Reuse Repository' can have the details of only perfect reusable elements. The remaining elements may be classified as 'Modification-Required' as per their 'Instability' value.

3.4 Format of Reuse Repository

Software Reuse Repository has the details of various reusable components of various products. Using Software Requirement Specification document reusable package or component may be selected. The components interface gives the services offered by a reusable component. To record and find the reuse elements the following data fields may be required in Reuse Repository.

Data Fields are:

1. Component Name
2. Coupling Type of Component
3. Stability Level of Component
4. Application Domain Name
5. Future Modification required or not
6. Quality Level of the Component
7. Programming Language

3.5 Selecting Components from Reuse Repository

To acquire an existing component from the repository the following things should verified [2].

1. Check the Functionalities of the component.
2. Check the Programming Language used for developing it.
3. Check component's interface and check the methods and input parameters
4. Memory size of the Component to check the performance of the system

4. RESULTS AND DISCUSSIONS

Here, the proposed methodology is a rigid-based classifying technique due to the 'Stability' metric of the selected software element. For an example, in 'Library Management System' the modules are 'Login', 'Registration', 'Update', 'Book Issue' and 'Book Return'. Among them 'Login' function point which needs 'user_name', 'password' and 'Bar-coded user _ID_card'. This component will do its task for 'User Authentication' with the help of Database. So it has only 'Control-Coupling' in the system to activate the remaining tasks in the system when an authorized user enters. As well as it does not require any data from other modules to initiate its duty. So it has only 'Export-Coupling' with no 'Import-Coupling'. Thus signals this 'Login' component to be 'Reusable' one in upcoming projects with a different database.

Table 2. Stability-Based approach for Library System

Component Name	Whether Import Coupling presence?	Whether Export Coupling presence?	Stability
Registration	yes	yes	In floating point
Book Issue	yes	yes	In floating point
Book Return	yes	yes	In floating point
Login	no	yes	1

From the example, 'Registration' component has both 'Import' and 'Export' coupling type then it may have 'Instability' value in floating point type. It means that it needs some more development effort to have all its clients elements into its space. That increases the size of the proposed component.

To implement a system the existing elements in reuse repository will be scanned to have the suitable elements. When suitable elements identified and used without modification then it may improve the quality of the system and work [6]. If number of reusable components increased then it many improve the quality of the system development and quality of the system.
That can be stated like

i) Reusable Component α Improving Quality

For an example, suppose selected 'Reusable Component' can do its service without any trouble then it may not need any modification. So simply it may connected with new identified modules in the new system to accomplish the task of system.
The following mathematical representation states that if number of reuse components are increased for a system it may reduce the cost of the total system development and increase the productivity of development team [11].

ii) Development Cost α 1 / No. of Reusable Component

For an example, suppose 5 components are identified in a system to implement the user's needs. Among them already two components are built in available in 'Reuse Repository', then development cost only needed for constructing other 3 components only. So it having an indirect proportional to Product Development Cost.

The below mathematical representation states that if a component is stable then it may not affect the improvement of the system quality and it may reduce the stress on 'Resource Management', 'Quality Management' and 'Time Management. If components are stable so it may not require any maintenance cost in future. So the development team can do further projects without having any stress. Because of that the outcome of products from development team will be increased [7].

iv) Stability α Improving Productivity

So the proposed methodology 'Stability based Reuse Clustering' can keep reuse components' detail in an effective manner and thus it may reduce the time for further discussion on the component to reuse in the proposed system development.

5. CONCLUSION

In classifying and acquiring reuse system elements lot of tools available [12]. Apart from application name and services of the components the stability of the component(s) is needed to track that component for further use in project development. Stability is an essential factor to represent the kind of dependency among components and communication among the components and their interior elements. So by applying 'Stability based Reuse Component Repository' can help the total system development with higher productivity in a very short period.

6. REFERENCES

[1]. Andrea Capiluppi and Cornelia Boldyreff (2007), "CouplingPatterns in the Effective Reuse of Open Source Software", Proceedings of International Conference on Software Engineering '07, Vol. 1, pp. 9-9.

[2]. Anthony Finkelstein, Spanoudakis G. and Ryan .M, (1996),**"**Software Package Requirements and Procurement", Proceedings of the 8th International Workshop, Vol. 8, pp.141-145.

[3]. Basili V.R, Bri L.C. and Thomas W.M (1994), **"**Domain Analysis for the Reuse Of Software Development Experiences", Experimental Software Engineering Group (ESEG), Vol. 11, pp. 86 - 95.

[4]. Jim-Min Lin (1997), "Cross-Platform Software Reuse by Functional Integration Approach", 21st International Computer Software and Applications Conference, Vol. 1, pp.402.

[5]. Rothenberger A.M, Dooley J.K, Kulkarni R.U and Nader Nada (2003),"Strategies for Software Reuse:A Principal Component Analysis of Reuse", IEEE Transactions on Software Engineering, Vol. 29, pp. 825 – 837.

[6]. Nancy Bazilchuk and Parastoo Mohagheghi (2005), "The Advantages of Reused Software Components", IEEE Transactions on Software Engineering, Vol. 23, No. 9, pp. 556-565.

[7]. Nunamaker, J.F., and Chen M. (1989), "Software productivity: a framework of study and An approach to reusable components", Proceedings of the 22nd Annual Hawaii international Conference, Vol. 2, pp. 959-968.

[8]. Parvinder Singh Sandhu, Janpreet Singh and Hardeep Singh, "Approaches for Categorization of Reusable Software Components", Journal of Computer Science 2006.

[9]. M. Halkidi , D. Spinellis, G. Tsatsaronis and M. Vazirgiannis,"Data mining in software engineering", Intelligent Data Analysis, 2011.

[10]. Man Deep Kaur, Parul Batra and Akhil Khare "Static analysis and run-time coupling metrics ", Oriental Journal of Computer Science & Technology, Vol. 3(1), 2010.

[11]. R.Kamalraj, B.G. Geetha, G.Singaravel "Reducing Efforts on Software Project Management using Software Package Reusability" , Advance Computing Conference (IACC 2009) IEEE International, Vol. 1. 2009.

[12]. Ali Hadian, Mahdi Nasiri, Behrouz Minaei-Bidgoli "Clustering Based Multi-Objective Rule Mining using Genetic Algorithm" International Journal of Digital Content Technology and its Applications, Volume 4, Number 1, February 2010.