

Expert System for Software Reusability

Ali Zeinulabdeen Alfakhry

Software engineering Department, Computer and
Mathematics College, Mosul University

Safwan Omar Hasoon, PhD

Software Engineering Department, Computer and
Mathematics College, Mosul University

ABSTRACT

In modern science, software engineering dealing with improving systems of the software within a short time to enhance the development process and reduce the cost as much as possible. Software reusability is the process that uses existing software components for implementing and updating software systems. Software engineers utilise reusability to reduce effort, time and cost in addition to increase the quality and productivity of software systems, also to check if software components can be reusability. Reusability of any component is computed depending on several metrics. In this paper, cohesion and coupling metrics are used to measure the reuse of software components of Object Oriented procedure (Java codes). The proposed expert system was constructed for solving software reusability problems. The expert system will be used as an intelligent technique to produce smart advice for software reusability problems.

Keywords

Software reusability, coupling, cohesion, expert system, knowledge base

1. INTRODUCTION

Software engineering stages dealing with the reusability of software with decreased effort and enhance development process. Reusability is the ability to use all or most parts of the same code or design in another software of applications [7]. Software reuse means constructing new software systems from existing software components. It has huge impact on productivity. Software elements that can be reused are software test cases, designs, prototypes, data, specification, plans, templates, frameworks and documents. Reusability of code is calculated by checking to what extent the code is reusable [6]. One benefit of reusability is that can be used to create new functionalities with small or localized code modifications when a change in implementation is required. Conformance to suitable design and coding standards increases a component's reusability. [7]

Good distribution of functions dealing with two key concepts of designing object-oriented: cohesion and coupling. Coupling is the expand to which the various subcomponents interact with each others. If highly interdependent between them, then changes in one of them have important effects on the behavior of others. Subsequently, loose coupling between subcomponents is a desirable characteristic of a component. Cohesion is the expand to which the functions or requirements performed by a subsystem are related, and it is an important software quality attribute, high cohesion is a characteristic of well-structured software design [2]. The functionality has been poor distribution to subcomponents when a subcomponent or one function is responsible for a number of not connected or unrelated functions [9]. One of the areas of artificial intelligence is expert system. An expert system (also known as knowledge based system) is a computer program that contains the analytical skills and knowledge of one or more human experts in a specific problem domain. The goal

of designing the expert system is to capture the knowledge of a human expert relative to specific domain and code it in way that the knowledge of an expert is available to a less experienced user.[13]

2. LITERATURE SURVEY

- Frakes and Terry (1996) – was the first person that proposed metrics and models on reusability of software. He suggested models based on cost benefits, assessing the maturity, the degree of reuse, the failure modes, and reuse library metrics. [12]
- Sushma Yadav, Dr. Sunil Sikka, Utpal Shrivastava (2014) - have proposed various coupling and cohesion metrics for OOP (Object Oriented Programming) software in "A Review of Object-Oriented Coupling and Cohesion Metrics".
- Neha Sadana, Surender Dhaliya, Manjot Singh Ahuja (2014) have proposed metrics to measure the reuse of software components for procedures oriented and object-oriented systems on the basis of coupling and cohesion in "A Metric for Assessing Reusability of Software Components".

3. SOFTWARE REUSABILITY

Reusability of Software is one of the important quality attributes in software engineering, which refers to "the extent to which a software module or work product can be used again in other applications in order to include new features". Industrial researchers suggest that about 20% of development costs can be saved by using reuse approach. Software reuse helps to reduce the development time and cost.[5]

In most engineering disciplines, systems are constructed by assembling exist sub systems that have been used previously in other systems.

Software engineering focused on original development previously, but it is now recognized that to achieve software that is more quick and at a lower cost in better form, we need to assume a design process that is based on systematic reusability of software.

3.1 Reusability benefits

Reuse of software component has many benefits as illustrated below :-

1. Increased dependability.
2. Reduced process risk.
3. Using specialists in effective manner instead of specialists of application that doing the same work on other projects, these specialists can improve reusable software that contain their knowledge.
4. Compliance standards.
5. Accelerated development. [8]

4. REUSABILITY METRICS

Two metrics of reusability are used in this paper, cohesion and coupling.

4.1 Cohesion Metric

Cohesion describes the intra-dependability inside components within a class [1]. Cohesion of one component is the degree to which it's responsibilities form a significant unit, high cohesion is necessary for good quality and reusability [9][10].

Simple Example about class with High Cohesion :

```

Class1
{
private Element1 ;
private Element2;
method1()
{
    uses Element1;
    uses Element2;
}
method2()
{
    uses Element1;
    uses Element2;
}
}
    
```

4.1.1 Cohesion metric used

Different factors that affect cohesion has been analyzed to measure cohesion metric Most effective factors are included in metric :

$$Chc = RSAc + RMNVc - RMVOC - RGVC \dots(1) [6]$$

Where : c = 1 to n (number of classes)

Chc = Cohesion Metric.

RSaC = number of shared attributes in class c / number of all attributes in class c

RMNVc = number of functions in class c returning no value to other classes / total functions in class c or function c returning no values to the other functions.

RMVOC = number of functions returning value out of the class c / total functions in class c or function c return value to other functions.

RGVc = number of public variables in class c / number of variables in the class c

In cohesion metric used , cohesion result is between : $-2 < Chi < +2$.

4.1.2 Benefits of high cohesion:

There are many benefits of getting high cohesion , some of these benefits are :-

1. Easy to understand modules/components .
2. System is easy to maintainability , changes in one component do not require any changes in other components.
3. Easy to reuse a component .[4]

4.2 Coupling Metric

Coupling describes the amount of inter-dependability among classes , it refer to the strength of program units [9] [10] .Program units which have high coupling are dependent on each other , loosely coupled program units are almost independent . low coupling is necessary for high quality and reusability of software [1]. Figure1 shown a class with high coupling :-

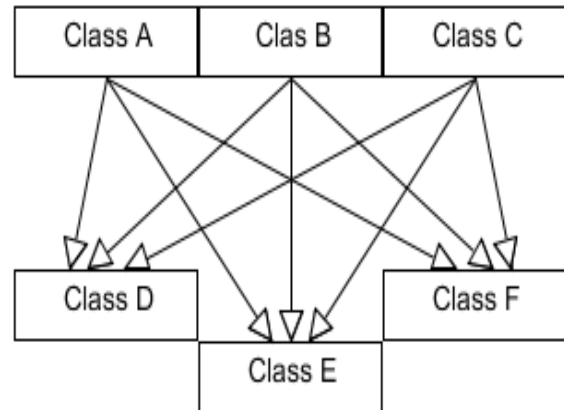


Figure 1 : Class with Low Coupling

4.2.1 Coupling metric used

Different factors that affect coupling has been analyzed to measure coupling metric between two classes . Most effective factors are included in metric :

$$COc = RSVc + RSFc + ICc \dots (2) \dots [6]$$

Where : c = 1 to n (number of classes)

COc = Coupling Metric.

RSVc = number of shared variable in class c / number of all variables in class c

RSFc = number of shared functions or public functions in class c / number of functions in class c

ICc = if Cc not inherits Ci then 0 else 1

In coupling metric used , coupling result is between : $0 < COi < 3$

4.2.2 Drawbacks of high coupling

There are many drawbacks of getting high coupling , some of these drawbacks are :-

1. Change in one class services a multiplier effect of changes in other classes.
2. Assembly of the classes might need more time and effort in order to the increased inter-class dependency.
3. A specific class might be hard to reusability and testing because dependent classes must be included.[4]

5. EXPERT SYSTEM

Expert system (Rule-based system) is a branch of AI (Artificial Intelligence). In computer science , a Knowledge base is the heart of expert system that include a set of rules " if condition then action " and facts to solve a specific problem . In software development , expert systems can be used to construct software that will provide an advice

on solving problem like a human expert . These systems also called a rule-based system.[14]

5.1 Expert system architecture

A typical expert system (rule based system) architecture consists of:

- knowledge base .
- working memory (for the current data) .
- inference engine , including two types that can be explained below :
 - 1- forward chaining (inductive, data driven) .
 - 2- backward chaining (deductive, goal driven) .
- user interface (possibly a Natural Language Interface NLI, menu, windows, etc) .
- explanation part (Why /How) . [11]

An architecture of expert system is shown in Figure2 :-

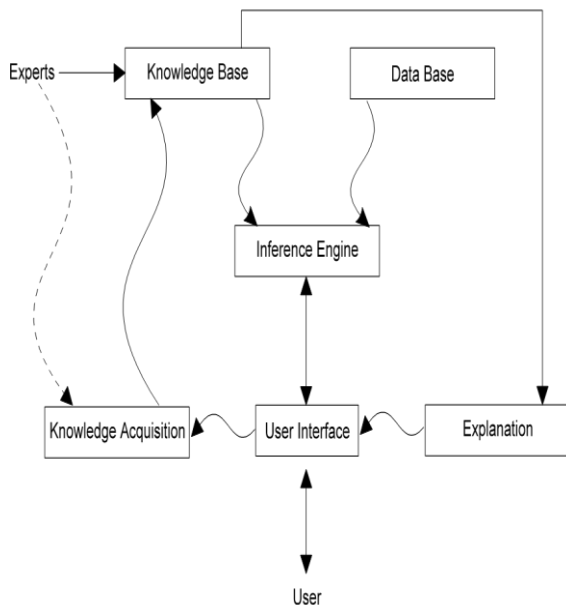


Figure 2 : Typical expert system architecture

5.2 Expert system benefits

Expert system introduced many benefits including the following :

- 1- Relatively inexpensive.
- 2- Do not require training .

6. PROPOSED EXPERT SYSTEM

The expert system is proposed for measuring the reusability degree of an object-oriented procedure (Java code) , as well as to make decisions about reusability status (i.e software can be reuse or not) depending on specific metrics (cohesion and coupling) .

Scanner and parser are applied on Java code for splitting it into set of classes , equation (1) and (2) are computed for each class , then proposed expert system was constructed depending on the results value of these equations.

6.1 Parts of expert system proposed

Proposed expert system architecture is illustrated and shown in figure (3) :

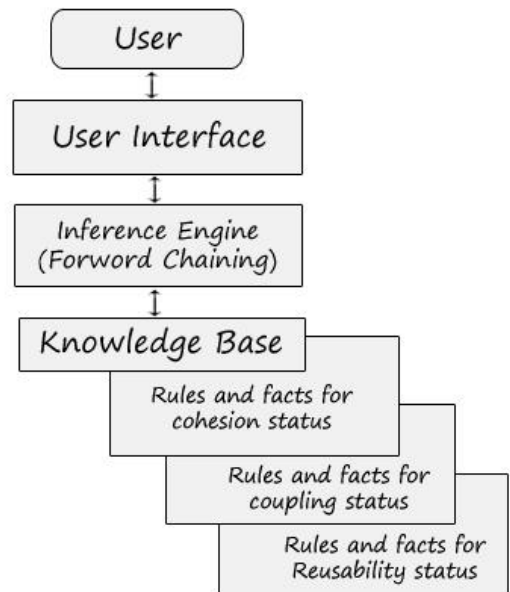


Figure 3 : Proposed expert system architecture.

The parts of the proposed expert system are described briefly below :-

- 1- User : represent the developer who will use the proposed application for computing software reusability .
- 2- User interface : an interface that have been constructed for reading java code and computing cohesion, coupling , and reusability status. Figure (4) illustrate the main interface which contains :

- *Select Java File* button : used for selecting java code file .
- *Run* button : used for executing the application .
- *Exit* button : used for closing the application.



Figure 4 : Main user interface.

The results of coupling values and status for each class from a selected java file when "Compute Coupling Status " button was activated are shown in Figure(5) below :-

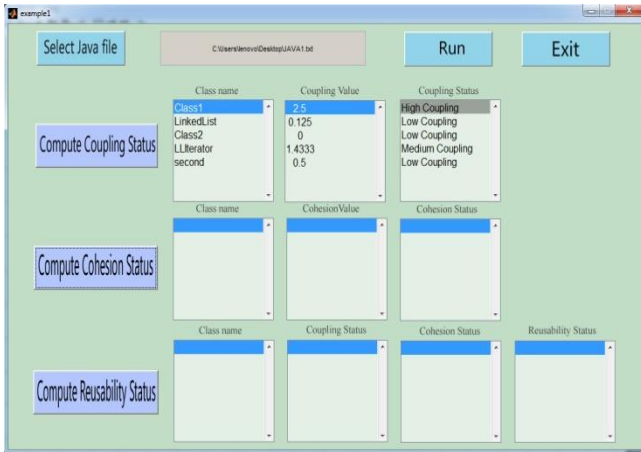


Figure 5 : Computing coupling status.

The results of cohesion values and status for each class from a selected java file when "Compute Cohesion Status" button was activated are shown in Figure(6) below :-

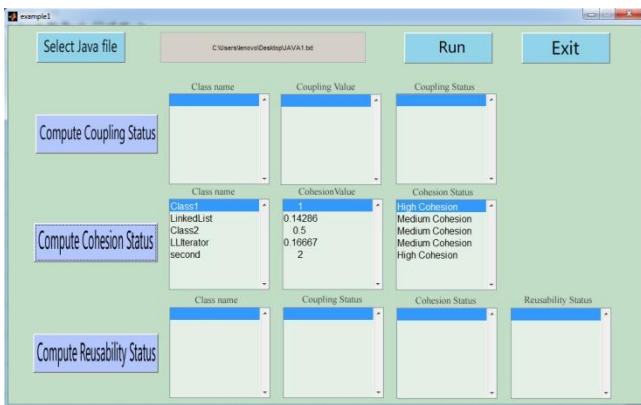


Figure 6 : Computing cohesion status.

The results of cohesion ,coupling and Reusability status for each class from a selected java file when "Compute Reusability Status" button was activated are shown in Figure (7) below :-

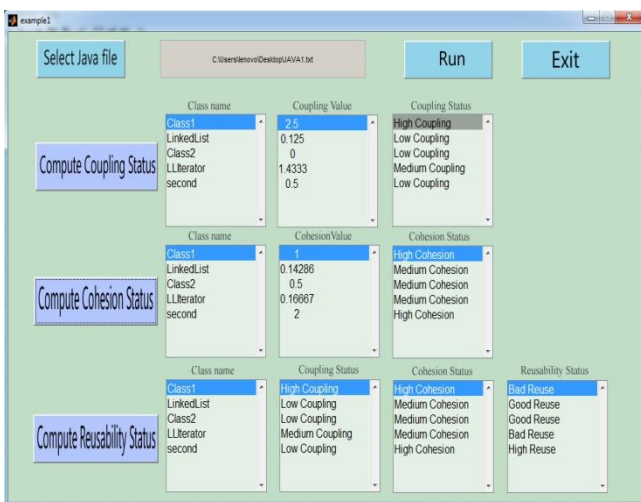


Figure 7 : Computing coupling , cohesion and reusability status.

3- Inference engine :- The forward chaining strategy is used through execute several rules and facts in knowledge base to make decision about reusability status .Many search

methodologies are used to get the advantages of their properties to obtain an optimal path to get desired goal .In this work , the breadth first search technique used as shown in figure(8) .

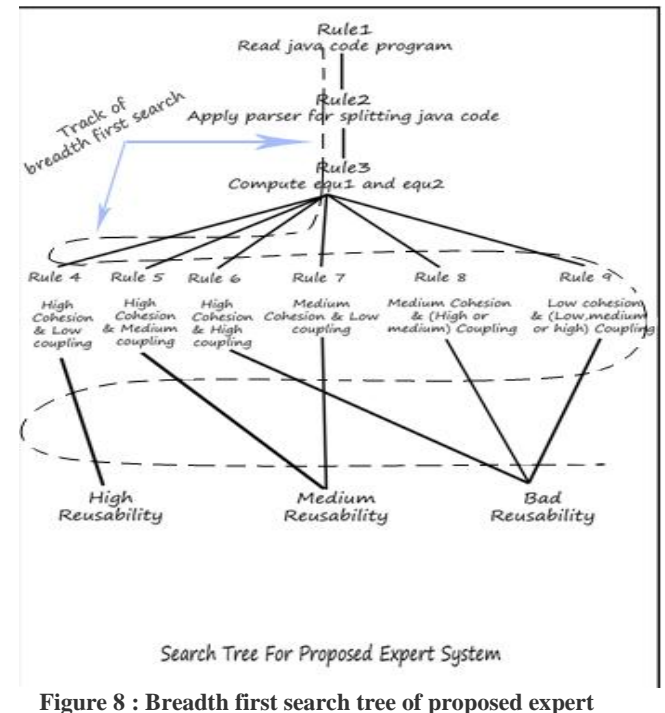


Figure 8 : Breadth first search tree of proposed expert system .

4-Knowledge base :- contains set of facts and rules as described below :

- Knowledge base of cohesion status is constructed depending on the result of equation (1) as shown in table 1 .

Table 1 : Knowledge base of cohesion status.

No.	Condition	Action
1.	If cohesion result is between (1 , 2).	Then High Cohesion
2.	If cohesion result is (greater than 0 and lower than 1).	Then Medium Cohesion
3.	If cohesion result is (greater or equal than -2 and lower or equal than 0)	Then Low Cohesion

- Knowledge base of coupling status is constructed depending on the result of equation (2) as shown in table 2 .

Table 2 : Knowledge base of coupling status.

No.	Condition	Action
1.	If coupling result is (greater than 2 and lower or equal than 3)	Then High coupling.
2.	If coupling result is (greater or equal than 1	Then Medium coupling

	and lower or equal than 2)	
3.	If coupling result is (greater than or equal 0 and lower than 1)	Then Low coupling

- Knowledge base of reusability status is constructed depending on status of cohesion and coupling as summarized in Table3.

Table3 : Knowledge base of Reusability status.

No.	Condition	Action
1.	If High cohesion , low coupling	Then High Reusability
2.	If High cohesion , medium coupling	Then Good Reusability
3.	If Medium cohesion , low coupling	Then Good Reusability
4.	If High Cohesion , high coupling	Then Bad Reusability
5.	If Medium Cohesion , high coupling	Then Bad Reusability
6.	If Medium Cohesion ,medium coupling	Then Bad Reusability
7.	If Low Cohesion ,low coupling	Then Bad Reusability
8.	If Low Cohesion ,medium coupling	Then Bad Reusability
9.	If Low Cohesion ,high coupling	Then Bad Reusability

6.2 Mechanism of the expert system

Decision making for reusability status is achieved by many steps as shown in Figure (9).

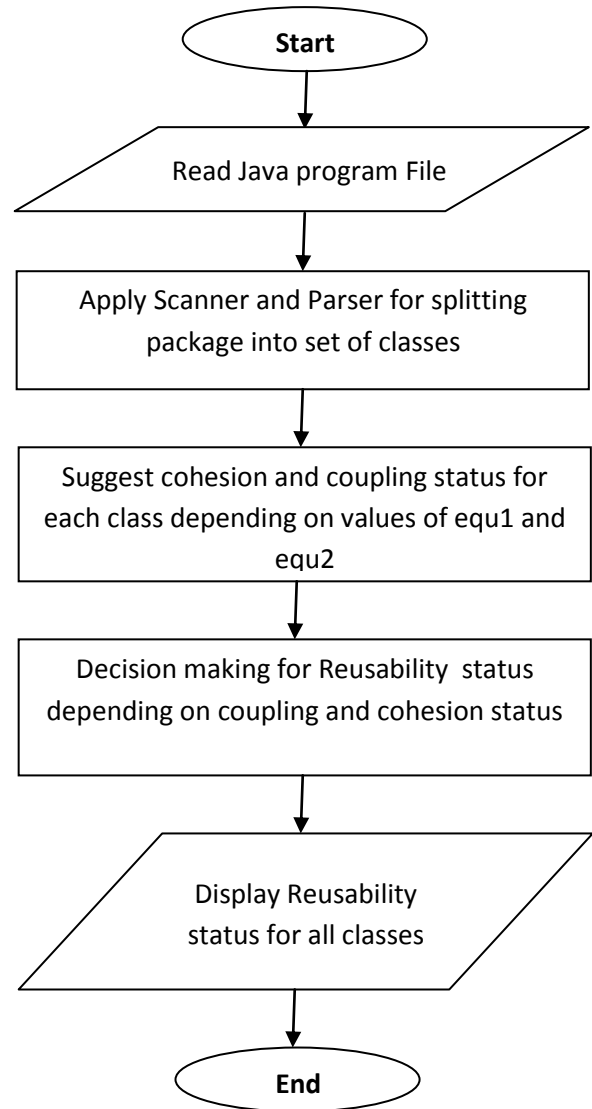


Figure 9 : flowchart of proposed expert system .

Computing coupling status is illustrate in figure (10) .

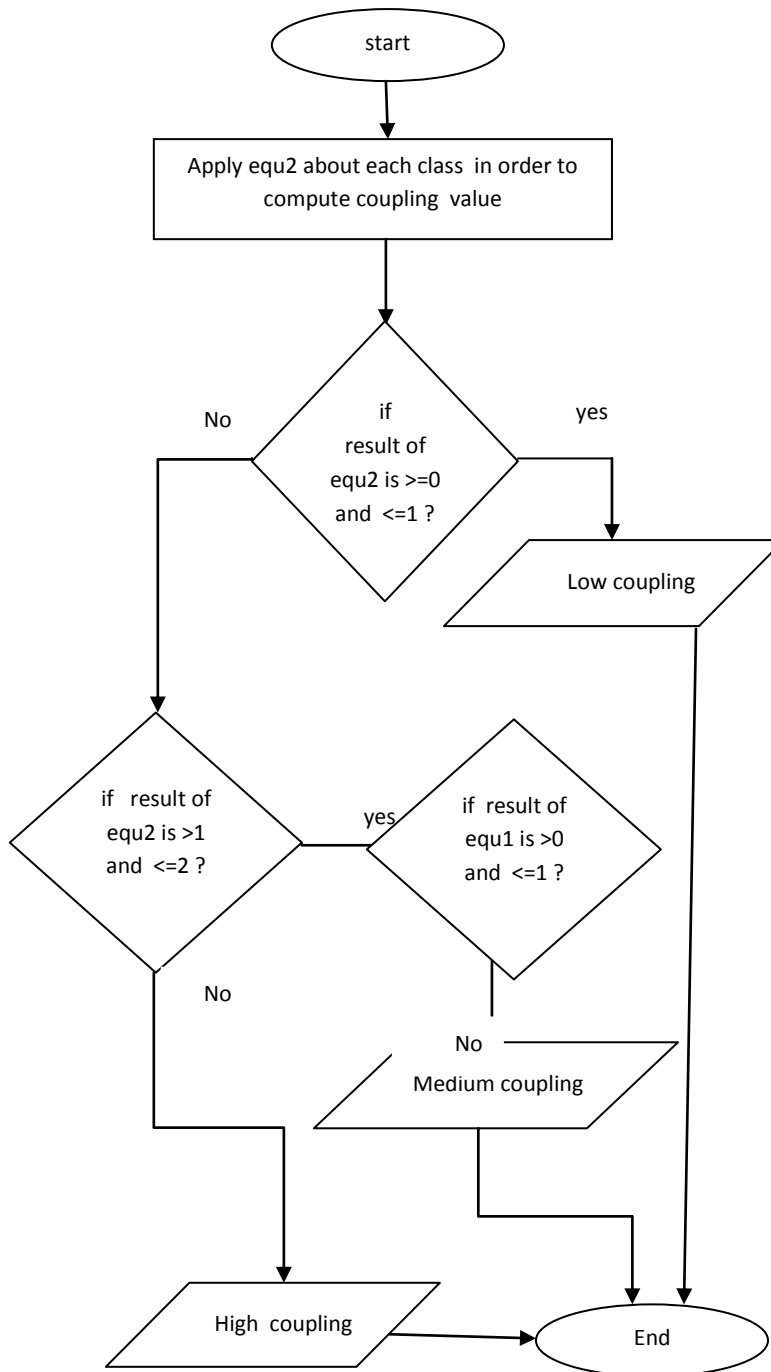


Figure 10 : flowchart of computing coupling status.

Figure(11) illustrate computing of cohesion status.

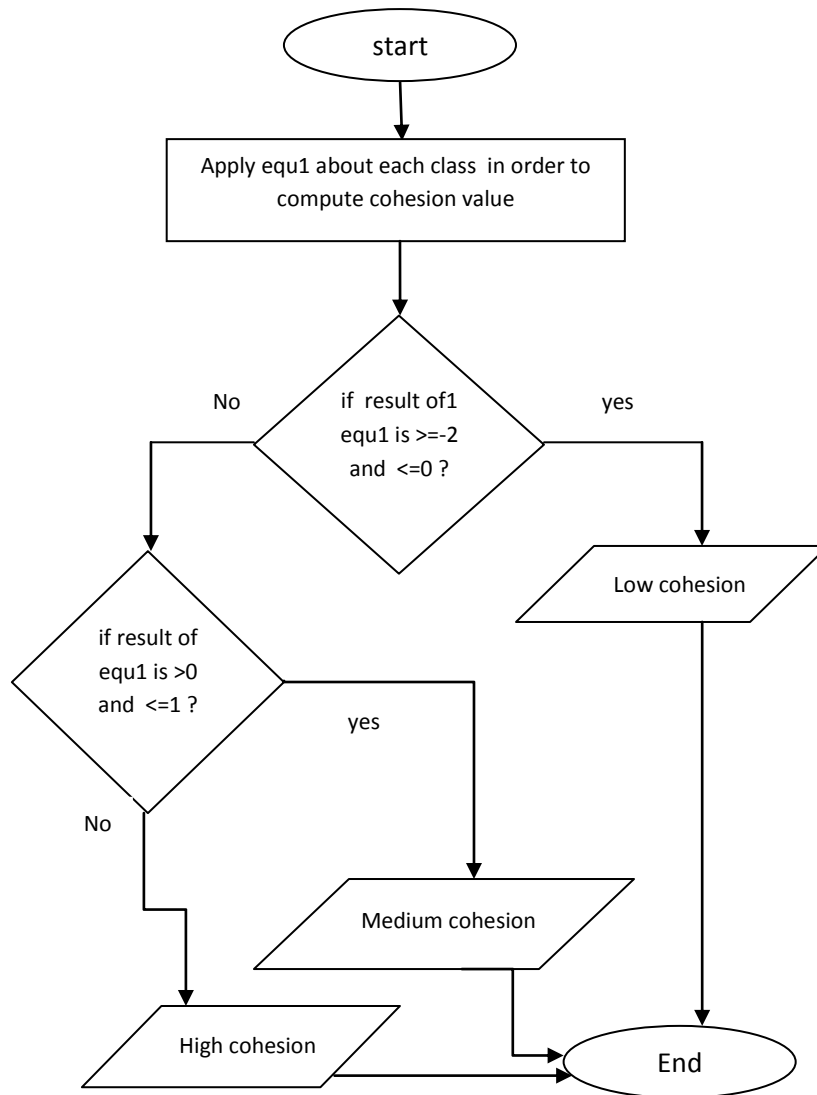


Figure 11 : flowchart of computing cohesion status.

After computing cohesion and coupling status, decision making for reusability status is done as illustrated in figure (12).

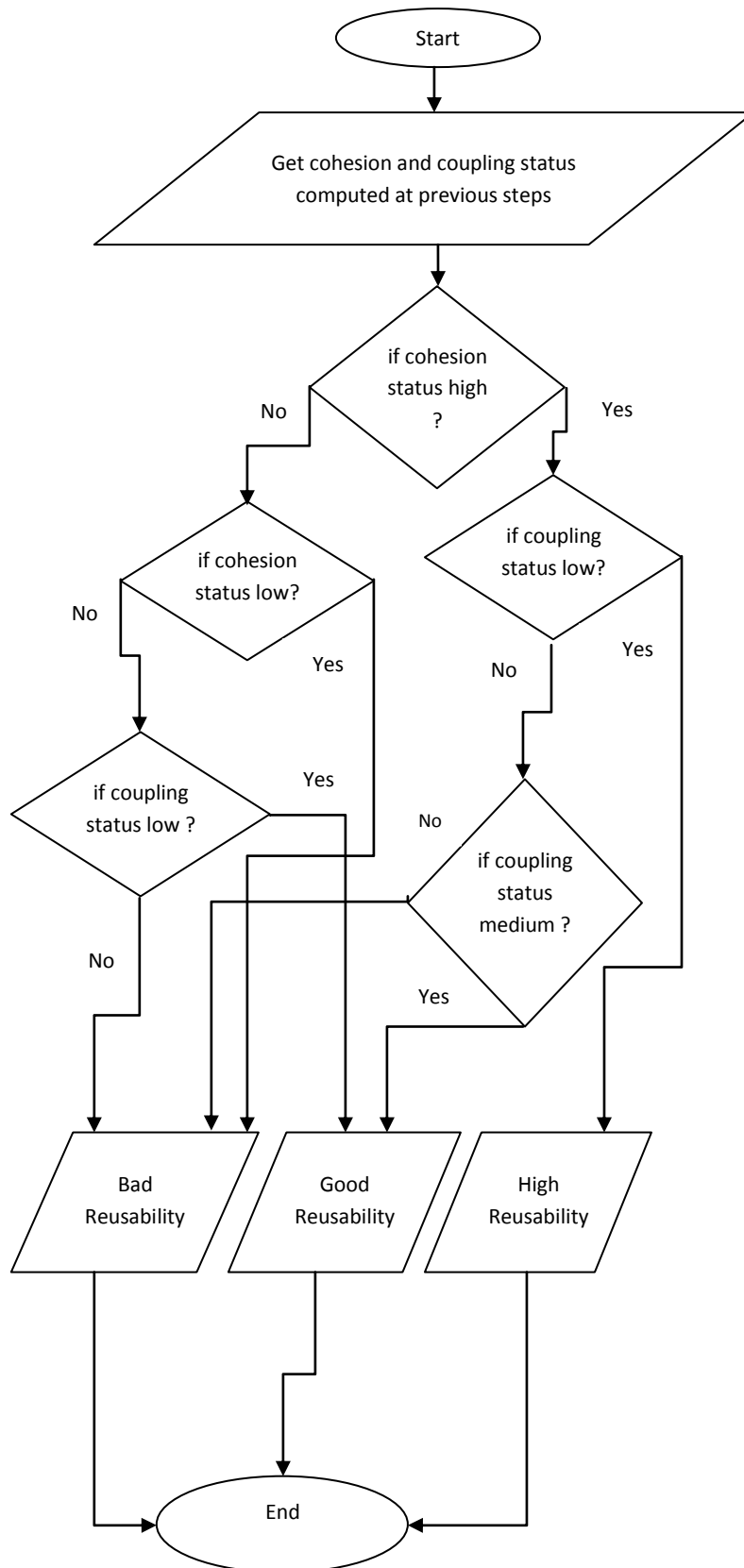


Figure 12 : flowchart of decision making for reusability status.

7. RESULTS AND DISCUSSION

In the proposed expert system , five classes were examined by calculating the values of coupling and cohesion to find their levels as shown in Table 4 and Table5 below :-

Table4: Coupling value and status for classes.

Class Name	Coupling value	Coupling status
Class1	2.500000	High coupling
LinkedList	0.125000	Low coupling
Class2	0.000000	Low coupling
LLIterator	1.433333	Medium coupling
Second	0.500000	Low coupling

Table5: Cohesion value and status for classes.

Class Name	Cohesion value	Cohesion status
Class1	1.000000	High cohesion
LinkedList	0.142857	Medium cohesion
Class2	0.500000	Medium cohesion
LLIterator	0.166667	Medium cohesion
Second	2.000000	High cohesion

Finally , reusability levels were determined based on levels of cohesion and coupling as shown in Table 6 below :-

Table6: Coupling, Cohesion , and Reusability status for classes.

Class name	Cohesion status	Coupling status	Reusability status
Class1	High	High	Bad reuse
LinkedList	Medium	Low	Good reuse
Class2	Medium	Low	Good reuse
LLIterator	Medium	Medium	Bad reuse
Second	High	Low	High reuse

As noted , Coupling values of class1 is 2.5 (high coupling) and cohesion value is 1 (high cohesion) , for this reason reusability level is Bad Reuse .

In second class LinkedList Coupling values is 0.12 (low coupling) and cohesion value is 0.142857 (medium cohesion) , which means reusability level is Good Reuse .

In third class class2 Coupling values is 0 (low coupling) and cohesion value is 0.5 (medium cohesion) , which means reusability level is Good Reuse .

In fourth class LLIterator Coupling values is 1.433333 (medium coupling) and cohesion value is 1.66667 (medium cohesion) , which means reusability level is Bad Reuse.

In fifth class second Coupling values is 0.5 (low coupling) and cohesion value is 2 (high cohesion) , which means reusability level is High Reuse .

8. CONCLUSION AND FUTURE WORKS

The proposed expert system was designed based on an intelligent technique in the software reusability field by constructing a knowledge base which includes rules and facts

that describe coupling and cohesion metrics . The system offers a smart decision about software reusability which reduce time and effort compared with traditional work . The proposed system proved to be good for recognizing whether specific codes can be reusable or not , and for optimizing any code to be reusable by increasing cohesion values or decreasing values of coupling depending on specific metrics. For developing this work , we suggest some future works as explained below :

- It is possible to use other metrics with cohesion and coupling metrics for computing reusability such as : complexity , flexibility metrics .
- Develop method to suggest improvements on any code that it is not reusable for making it reusable code

9. REFERENCES

- [1] Annushri Sethi , Prof. Ritu Tandon ." A Novel Approach to Find Reusability using Coupling and Cohesion Metrics ". Feb-2017 .
- [2] Habib Izadkhah, Maryam Hooshyar , "Class Cohesion Metrics for Software Engineering: A Critical Review " ,vol.25, no.1(73), 2017.
- [3] Karminder Kaur , Gurjot Kaur. "Component Reusability of a Software System based on Cohesion and Coupling " . July 2016.
- [4] N. Rajkumar, C. Viji1 and S. Duraisamy ." MEASURING COHESION AND COUPLING IN OBJECT ORIENTED SYSTEM USING JAVA REFLECTION " . APRIL 2015.
- [5] Charu Singh , Amrendra Pratap, Abhishek Singhal "An Estimation of Software Reusability using Fuzzy Logic Technique" ,In IEEE 2014.
- [6] Neha Sadana, Surender Dhैया, Manjot Singh Ahuja , "A Metric for Assessing Reusability of Software Components ",Issue 4, Volume 1 ,February 2014 .
- [7] Ms.Kirandeep Kaur , Ms.Rekha Rani , Ms.Jagdeep kaur ."Code Reuse & Reusability of the Software " . Vol .2 April 2013.
- [8] Roger S. Pressman, Ph.D. "Software Engineering, 7th edition " . Chapter 18 , 2010.
- [9] Guigui, Paul D. Scott."Measuring software component reusability by coupling and cohesion," In journal of computers, Vol.4, No. 9, September 2009.
- [10] Kavitha , Dr. A. Shanmugam ."Dynamic coupling measurement of object oriented software using trace events," In IEEE Transaction, 2008.
- [11] N. Kasabov, "Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering " , MIT Press, 1996.
- [12] William Frakes, Carol Terry, "Software Reuse: metrics and models", ACM Computing Surveys (1996).
- [13]Dennis Ritchi, Artificial intelligence, Tata McGrawHill, New Delhi (1996).
- [14] John Durkin , "Reports on Over 2500 Developed Expert Systems " , 1993.