Volume 3, Issue 5, May 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Code Reusability Model for Object Oriented Software Design

Mayank Mandloi
Scholar –M.Tech(IT)
Department of IT
PCST, Indore India

Prof. Sachin Patel
Head of Dept. IT
Department of IT
PCST, Indore India

Prof. Rakesh Pandit
Department of IT
PCST, Indore
India

Abstract— In daily routines now in these days we use various kinds of applications and utilities which are based on computer. To make any application useful a large team is work to provide the services and various development methods that work in background. In software industries various efforts are made to reduce the cost and time of development any application, in this project we work to improve the development cost and time by using the previously written code utilization. Our concept is based on the code reusability and analysis in object oriented programming. To develop such kind of methodology we propose, design and implement a code analysis tool by which developer analyse how much amount of code is reusable in newly development requirement. The performance of the proposed model is evaluated in terms of accuracy of filter the amount of code accurately classified for future use. This is derived using the user's relevance feedback.

Keywords— software designing, code reusability, object oriented programming, life cycle, user feedback.

I. INTRODUCTION

In this era of technology each and every day in our life interacted with the computers and their complex applications, these applications are help us to perform complex task in few minutes. this fast and easy computation ability is developed using a lot of efforts and work force of any software industry thus the complexity of any task is given in the manner of time and human resources are applied over the particular task to solve them. The software development requires a large amount of efforts and a good plan to work for. This planning and implementation involve various and different kinds of task for provide the proper and best solution. Thus here we start from the software life cycle. Software development life cycle is also known as SDLC. SDLC is a combination of different activity, planning, calculations and team efforts. that is actually a possible steps to achieve optimum solution for a complex problem space. There are various different models are available for SDLC processes, most of the designers change the steps of processing, some authors apply only five steps in SDLC and some of them are six. In a SDLC model each processes or step describing approaches to a variety of tasks or activities that take place during the development process. Some people consider a life-cycle model a more general term and a software development process a more specific term. Most of the software development process models are contains some common steps or we called the activities these are:

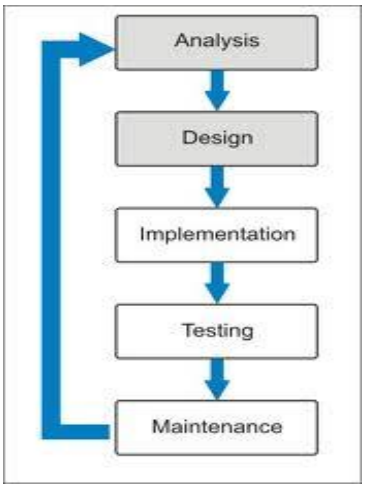


Figure 1 software development life cycle

To develop software first required to make **Analysis** of the given problem is take place in first after that developers team work with client to get knowledge the basic need and different **requirements** of client. Then that is planed over paper and different software engineering **design** tools, after that all work are written using code that is appropriate to solve problem that process is known as **implementation**. After that is cross checked or tested over different parameters to

provide quality software. Finally it is deployed over client end. in this section we provide the general introduction of working domain, in the next section we provide the problem identification of the studying domain, and solution obtained, in addition to here we provide the implementation of our proposed work and results obtained by us. Finally we conclude the complete designed system.

II. BACKGROUND

To find exact problem and their solution we study the different papers and articles that are previously developed for the concept of code reusability, with the different ways and over different methodologies. In the software industries the main goal is to save the time and the human resources over different task in suitable manner. For the better utilization of resources required to make efforts to manage them.

Code reusability is the probability a part of source code that can be used another time to add new functionalities with slight or no adjustment. Reusable modules and classes reduce implementation time, increase the possibility that prior testing and use has removed bugs and localizes code modifications when a change in execution is required.

Reusability indicates some open supervision of build, packaging, distribution, installation, configuration, deployment, and maintenance and upgrade issues. If these issues are not considered, software may appear to be reusable from design point of view, but will not be reused in practice.

Software reuse is the most promising approach for increasing efficiency and improving the quality of software in the software industry. It is simple in concept; successful software reuse is difficult in practice. A reason put forward for this is the dependence of software reuse on the situation in which it is implemented. Some problematic issues that lead to be addressed related to software reuse are given below:

- A clear and well-defined product vision is an essential foundation to an SPL.
- An evolutionary implementation strategy would be a more pragmatic strategy for the company.
- There exists a need for continuous management support and leadership to ensure success.
- An appropriate organizational structure is needed to support SPL engineering.
- The change of mindset from a project-centric company to a product-oriented company is essential.

Most of the IT Companies are works on object oriented programming environment (OOP). In OOPs, it is possible to use certain code again for different modules through inheritance. Some companies are work over component based models. And in both of the cases code reusability is a profitable issue.

Reuse models and metrics are categorized into types: (1) reuse cost benefits models, (2) maturity assessment, (3) amount of reuse, (4) failure modes, (5) reusability, and (6) reuse library metrics. Reuse cost-benefits models include economic cost/benefit analysis as well as quality and productivity payoff. Maturity assessment models categorize reuse programs by how advanced they are in implementing systematic reuse. Amount of reuse metrics are used to assess and monitor a reuse improvement effort by tracking percentages of reuse for life cycle objects. Failure modes analysis is used to identify and order the impediments to reuse in a given organization. Reusability metrics indicate the likelihood that an artefact is reusable. Reuse library metrics are used to manage and track usage of a reuse repository. Organizations often encounter the need for these metrics and models in the order presented.

1. architectures	6. estimates (templates)
2. source code	7. human interfaces
3. data	8. plans
4. designs	9. requirements
5. documentation	10. test cases

Reusable Aspects of Software Projects

1. Inheritance concept in OOPS: Inheritance concept is the most important property of OOPS. It is a technique that creates a new class from an already defined class. The new class contains all the attributes of the old class [1][9] in addition to some of its own attributes. Additionally it can override some of the attributes and features of old class. When there is a need for some functionality, user can inherit those related functions of the base class and use it. Once the class is defined then it can also be reused several times by other applications after being inherited into the class which suits that particular application. Through inheritance all the applications can be made to inherit the designed class into a new class and can be used in new class. The object class should be at the zenith of the class hierarchy. Every class should descend from it in a direct or indirect manner. In general the derived class inherits some or all of the traits from the base class and a class may inherit properties of more than one class in a single or more than one level [2].

2. Segment dependencies: We classify segment dependencies in two ways: contract or noncontract, and explicit, implicit, or informal. Contract dependencies [3] are those that have been intentionally introduced by the programmer, this dependency sometimes voluntarily required for the code segment. Whereas non-contract dependencies are taken accidentally, Code that relies on non-contract dependencies is less likely to be reusable. Encapsulation [2][9] can be seen as reducing such dependencies in object oriented programs. Explicit dependencies are those that are described directly in the language. Implicit dependencies are those for which there is no language support for describing them, but which can nevertheless be checked in some way or other. Informal dependencies cannot be described in the language, nor can they be checked. Informal dependencies are not as helpful as implicit dependencies because there is no way to ensure they have been met. Implicit dependencies are not as helpful as explicit dependencies because, it is not obvious what must be done to meet them.

III. PROPOSED WORK

Due to large domain of software industry and engineering each and every designed system contains different functionality and use. Thus each time new code and functionality added and a large amount of effort are required due to a sort time. To save consumption of efforts and provide the ease to development team new systems required by which the efforts required to write complete code is reduced. To overcome this kind of problem code reuse from previously written code chunks is a beneficial way to save the time and efforts.

Due to study we found that previously written model and tools are just returns the reusability factors in terms of functions and components thus required a new semi-automatic code analyser by which the percentage of code part is returned. And also extract or point out the code where it resides in code model.

To resolve the above described problem we propose a new semiautomatic feedback based code analyser that helps us to review how much code lines are reused and where the code is written in old code.

To derive solution we decompose our complete task in below given steps

1. **Input source code for analysis:** in this part of system processing we select the old written java code directory as input.
2. **Class listing:** here we read complete directory of source code and list all classes in the code.
3. **Member function listing:** in this step system select individual class and discover all member functions and attributes.
4. **Feedback matrix:** here developer add the requirement of new system as per given format of system.
5. **Search for code:** here system reanalyse complete member function list and list best matched function.
6. **List of code chunks:** here developer again adds feedback to filter discovered results and here we got final list of functions and classes which is used in new system.

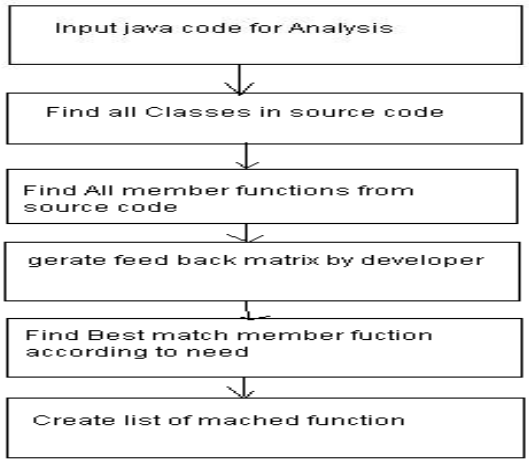


Fig 2 shows the basic flow of proposed work

IV. IMPLEMENTATION

The proposed system is implemented through the java framework for and using the net beans IDE. Java is an object oriented programming language and uses a rich class and library for implementation. Additionally this IDE enable to use different other external codes and library.

In this section includes the classes written, libraries used functions and methods which is used. These classes are some important classes that are used methods and functions.

User defined classes

S. No.	Class Name	Description
1	AddClass	This class is used to add new class into the code repository for future use
2	Frm_Query	At the time of code search this class is used as user interface for make search
3	frmGraph	This class is used to show the system performance in graphical representation
4	MyClass	This class is used to read the code file and provide the way to extract different components of the class
5	Parameter	This class is used to identified the methods and functions return type and data types that

		a method can return or accept
6	Method	This class is used to omit the methods definition found after query in database

System class library:

S. No	Class Name	Description
1	java.io.File	This class represents a file name of the host file system. It must use the file name conventions of the host platform. The intention is to provide an abstraction that deals with most of the system-dependent file name features.
2	java.sql.*	Used to get methods and classes for JDBC connectivity
3	java.util.ArrayList	Resizable - array implementation of the List interface. Implements all optional list operations, and permits all elements, including null. In addition to implementing the List interface, this class provides methods to manipulate the size of the array that is used internally to store the list.
4	java.util.Collection	The root interface in the collection hierarchy. A collection represents a group of objects, known as its elements. Some collections allow duplicate elements and others do not. Some are ordered and others unordered.
5	java.lang.*	the classes in the java.lang package are so essential, the java.lang package is implicitly imported by every Java source file. In other words, you can refer to all of the classes and interfaces in java.lang using their simple names.
6	javax.imageio.ImageIO	Class containing static convenience methods for locating Image Readers and Image Writers, and performing simple encoding and decoding.

Methods and signature

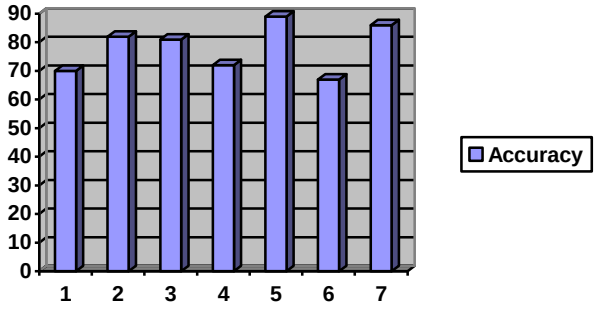
S. No	Method	Description
1	CreateFile()	This method used to create a new file using the file name and its contents
2	Analyse()	This method is user defined used to read all files and return

		different flags, variables and methods
3	getClassTokens()	This function is used to tokenize the complete class
4	getMethodInfo()	This class return the methods information
5	getFullFile()	The complete file is returned using this class
6	SearchMethod()	This function is used to search the method which is required by the programmer

V. RESULTS

In this section of the proposed work we provide the performance evaluation of the designed system. the performance of the system is given using the following parameters.

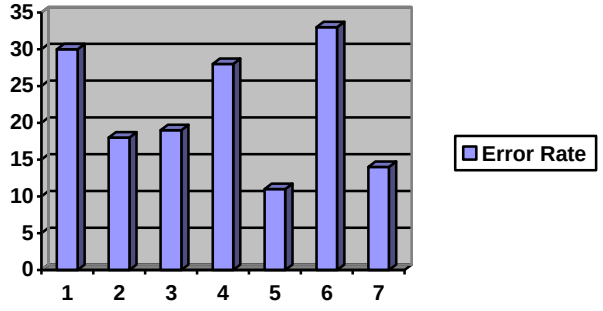
Accuracy: this parameter is providing the information how accurate results we obtained by the system after querying to the code repository. Accuracy of the system during different experimental scenarios are given as

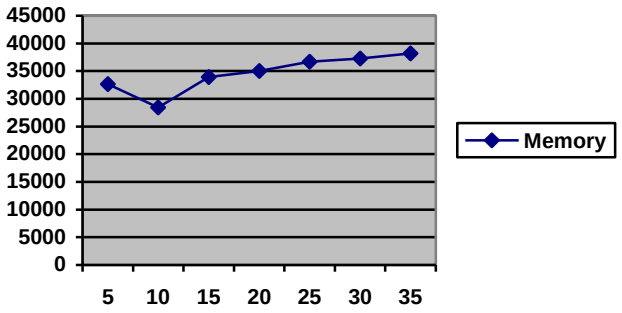


Most of the time system generates or searches much nearer results as the parameters are supplied in the query interface.

Error rate: this factor provides the difference between the actual positions of achievement and desired to achieve. That is defined as

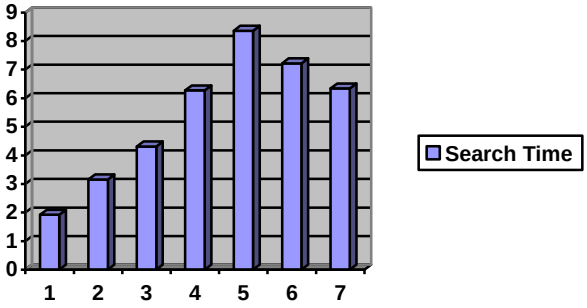
Error rate = 100 – accuracy





Memory uses: This parameter reflects the memory cost required to execute the system successfully. That is directly proportional to the size of code required to analyse, which is performed using different size of file and the observation results are given below.

Search Time: the search time is known as the time required finding the desired method using system.



VI. CONCLUSION AND FUTURE WORK

The proposed system which is required to implement is completed, additionally the performance evaluation of the system is completed and we found the following facts after implementation of the desired system.

developing more programmer friendly environment Microsoft corporation developed object based programming languages, where most of the user interfaces are provided as component, which is required expertize hands for configuring them and use them with the code this concept is sometimes also called the COM(component object model). But due to need of distributed applications and their relative resources again modified as the DCOM which is based on distributed component model where all the components are made available as classes.

but not all components are support the reusability thus we can say the component based models and their architectures are partially supports the reusability model actually here required to provide the method level abstraction for implementation purpose. here we introduces the code reusability for method level search where old methods are available as library for searching the most relevant codes form the code repository and provide ease to programmers by saving time to write the code and make logic for writing code.

References

[1] A Software Platform for Component Based RT-System Development: Open RTM-Aist, S. Carpin et al. (Eds.): SIMPAR 2008, LNAI 5325, pp. 83–94, 2008. c浼Springer-Ver lag Berlin Heidelberg 2008

[2] Engineering Web Applications for Reuse, 1070-986X/01/$10.00 © 2001 IEEE

[3] Exemplar: A Source Code Search Engine for Finding Highly Relevant Applications, IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 38, NO. 5, SEPTEMBER/OCTOBER 2012

[4] Comparative Study on Main and Sub-code Reusabilities, IJCSNS International Journal of Computer Science and Network Security, VOL.12 No.7, July 2012

[5] Software Metrics to Estimate Software Quality using Software Component Reusability, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 2, No 2, March 2012 ISSN (Online): 1694-0814 www.IJCSI.org

[6] Software Reconfigurable State-of-the-art Communication Suite for Fighter Aircraft, 2011 International Conference on Communication Systems and Network Technologies

[7] Refactoring for Multi-Dimensional Reusability, Innovative Systems Design and Engineering www.iiste.org ISSN 2222-1727 (Paper) ISSN 2222-2871 (Online) Vol 2, No 4, 2011

[8] MEASURING SOFTWARE REUSABILITY USING SVM BASED CLASSIFIER APPROACH, international Journal of Information Technology and Knowledge Management January-June 2012, Volume 5, No. 1, pp. 205-209

[9] Reusability of Software Components using J48 Decision Tree, International Conference on Artificial Intelligence and Embedded Systems (ICAIES'2012) July 15-16, 2012 Singapore

[10] DESIGNING CODE LEVEL REUSABLE SOFTWARE COMPONENTS, DOI: 10.5121/ijsea.2012.3116

[11] A Comparison of Development Effort Estimation Techniques for Web Hypermedia Applications, E Mendes, I Watson, C Triggs - . Eighth IEEE , 2002 - ieeexplore.ieee.org

[12] Applications of Fuzzy Logic to Software Metric Models for Development Effort Estimation, A Gray, S Macdonell - Society, 1997. NAFIPS'97., 1997 Annual, 1997 - ieeexplore.ieee.org

