An Object Oriented Reusable Framework to Support Novice Programmers

Alhassan Adamu, and Wan Mohd Nazmee Wan Zainon

Abstract—Object oriented programming is one of the “stalagmites” of computer science and is a useful skill that constitutes one of the nucleus competencies expected from every graduate of computer science. It prepares students with technical knowledge and essential skills needed to work in real life software projects. However, it requires a lot of cognitive thinking and problem solving skills to translates abstract though in to real life computer program. This is very hard to achieve during the period of one semester course. This paper proposes the reuse of chunks of source code without its associated artifacts by the novice programmers during the period of programming course. Source code reuse is becoming the most popular form of reuse because its effects on productivity are immediate.

Index Terms— Software Reuse, Novice Programmers, Query, Repository Model.

I. INTRODUCTION

Software reuse is the use of existing software components to create new software system. It consists of two main tasks: developed for reuse and developed by reuse. In develop for reuse software components are developed and stored in a component library for later use. While in develop by reuse; previously developed components are adapted to construct new software system [1, 2]. Reusing existing components to developed new software system hold a promising future in software engineering practice in enhancing software development process by reducing time to market, improving software quality, and effective of use of specialist [3].

There are many software components that could be reuse ranging from requirement specification, analysis, design, implementation, test cases, documentation and any intermediate knowledge found useful during the software development process. Source code reuse is the most common and easiest type of software reuse since its effect on productivity is immediate [4].

In recent time object oriented programming (OOP) has become the most influential type of programming paradigm both in academia and industry. In fact, the interest in teaching OOP to first year computer science student has increased in recent years, with almost every university computer science course contained object-orientation in its curriculum. Initially teaching OOP was seen as an advanced topic and was only taught to higher level students. Consequently, it is now realized that the OO concepts are a good basis for teaching fundamental programming skills to even the first year computer science students [5].

There are strong benefits in early teaching of OOP concepts to computer science students; some of which are: i. promotes well-structured programming practice. ii. Promotes reusability practice among students by reusing existing code to developed new program in addition to the development of new code iii. Teach students the important software development concepts such as “software reuse” at their early stage [6]. However, there are difficulties involve such as lack of tool supports, language selection, difficulty in finding reusable code, and difficulties in developing course projects and assignments [7].

Based on the aforementioned benefit of OOP, this paper proposes a reuse technique to assist novice programmers reuse OOP source code in the process of learning programming course. It’s expected that technique will reduce the student cognitive load while trying to learn how to program. The rest of the paper is organized as follows. Sections 2 discuss briefly on the novice programmers Section 3 discuss briefly on the related approaches. In Section 4 we present our approach. Finally, the paper concluded in Section 5.

II. NOVICE PROGRAMMERS

As main challenges to every novice is learning something new. Novice programmers are faced with challenges in learning the OOP concepts at the beginning. On the other hand experienced programmers are faced with the challenges of shifting from procedural programming paradigm in to the world of object-oriented [5]. Some literature regards novice as undergraduate students that has no prior industrial experience [8, 9]. In this paper novice programmers are referring to undergraduate students.

Study on novice programming skills has long history in computer science education research. For instance [10] performs a series of workshops on the empirical studies to investigate the additional knowledge that expert programmers have more than novice programmers. The study was based on the programming techniques using Pascal. Of recent [11] conduct an investigations on the most common programming mistakes among students. 250,000 students were taken from large Blackbox data set. The study investigates the most common errors among novice programmers and found that syntax errors are the most popular type of error among novice programmers. Syntax errors are errors that occur due to the violation of the programming rules. In [12] study in the difficulties in learning programming language by novice programmers was carried out to support the development of learning materials for basic programming course. The study...
found that, one of the biggest problems of novice programmers is not in understanding the basic concepts of programming but rather how to apply those concepts. Robinson [13] suggest that to ease the learning process by novice programmers a combine teaching approach of theory and practical should be employed. The Theory introduces the novice to the basic programming concepts while the practical learning helps them to learn the real life program implementation.

### III. RELATED APPROACH

Searching of existing source code from open repository such as stack overflow 1 has become a regular activity by most of novice programmers. For example, consider a novice want to create a “change password form” using Java. The novice turn to the online search to find the snippet code that will solve the problem using keyword search such as: “how to create a change password form in java”. The challenge is that the result in first page of the browser may not be relevant to the search criteria since the query may return millions of results. Research on the concept of assisting/reducing difficulties in learning programming language by novice programmers has been a long topic of discussion in computer education research. Many studies have developed a technique in C++ and Java programming to assist novice in learning how to program.

Lo. et al. [14] proposed gfpp (gap-filled programming practice) method to reduce the novice programmers cognitive load when learning how to program. The method is based on fill up the blank, in which instructors give a complete program structure and block out a part of the program to allow only the portion required to fill up by the students. In this way the students can focus on only the blank parts rather than a whole complete program. Zschaler et al. [15] use SALEPOINT2 in teaching object-oriented software development concepts to undergraduate. SALEPOINT is a java-based framework for creating business applications; it was jointly developed and maintained by Dresden and Munich University. Students are exposed to working with reusing existing framework to developed new applications and also introduce them into real life software project development. This help to equip the student on how to handle a more practical challenges.

In [16] object-oriented design framework was proposed to support novice software engineers during software design process. The framework consists of process and the components involved in the process. The process refers to the activities involved in the design stage while the components are the elements in the design process. The process in the framework involved finding of objects required for new system during analysis phase, followed by identification of the system classes by their common objects. Attributes (data) and method (behavior) of a class are determined next in order to identify the responsibility of a class. Finally, the framework identifies the relationship between the classes. Authors in [17] proposed and automatic debugging assistance (ADA) to help student finds bugs during programming assignment. The system was developed as an extension to Programming Assignment aSsessment System (PASS) developed by City University of Hong Kong for teaching and learning of programming skills [18]. With ADA students program output can be compared with expected outputs uploaded by the course instructor. Students are allowed to make submission and test their program for any number of times before deadline.

Our approach is an integrated reuse framework whereby students search for similar source code from within the repository of programming course. Students have the liberty to reuse existing code developed by other students during previous programming course. This approach reduce the difficulties for the student to go online looking for code snippet and also control the possible code plagiarism among the students.

### IV. UNITS

This section describes the proposed approach. depicts the proposed framework. Reuse process can be classified in to two possibilities: systematic or opportunistic. In systematic software reuse; components are purposely developed and stored in components repository for later use. This process requires additional resources such as time effort of which many organizations are unwilling to spend. On the other hand, the opportunistic is the easiest and direct type of reuse. In this process software developers only come to the conclusion that previous components is reusable when it can be in cooperated in to new system [19]. The second possibility may not require additional cost from the organization but it will require proper tools to assist the developers in finding and reusing such component. The work in this paper will utilize both the possibilities in the sense that the framework allows the student to purposely develop a component to be kept in the repository for future use as well as allow the student to reuse existing source code from repository. We subsequently discuss the functionality of the modules in the proposed framework.

#### A. IDE

IDE stand for the integrated development environment, it provides a standard facility to the programmers to write code during software development. The IDE in this framework allows the students to write a computer program in Java language, the environment will recognize all the reserved words in Java (e.g. class) in order to assist the student in writing correct java code. Moreover, the IDE allows the re-user to write a partial code and search for complete code via C_Extractor from repository for reuse as well as extract and store new reusable code for future reuse.

#### B. C_Extractor

This is main module of this framework it stands for code extractor. It is an intelligent module that maps user query to the repository codes for extraction. This module will read as input a source code file, processed the file using a strong .Net framework regular expression technique called regex. Regex is a powerful, flexible and efficient method for processing text. It provides an extensive pattern-matching notation of regular expressions to enable and efficient parse of large amount of

---

1. [www.stackoverflow.com](http://www.stackoverflow.com)
3. [https://msdn.microsoft.com](https://msdn.microsoft.com)
text to find specific character and pattern. It validates text to ensure it matches a predefined patterns defined by user.

C. Extractor performs two functions: i. Extract and store new source code in to repository for future use (developed for reuse) and ii. Map user query with repository code for retrieval (developed by reuse). During extraction all noise within the source code are remove to prepare the source code for extraction. Noise here is referring to text ignored by compiler during program compilation such as space, tab and comments. Comments are later sent back to code which will use as signature as well as descriptor of classes within the source code. Fig. 2 shows the context view of this module.

C. Reusability Test

Reusability test, is concerned about how a system or it is part could be used in a new system [20]. This module is added to the framework in order to measure the reusability of a source code before it is in-cooperated in other system as well as to measure the reusability of code before storing in to repository for future use. This module is important in order to avoid saving of redundant code to the repository. Redundant code does not add any value to the repository unless increases the access time during retrieval process, therefore such code are discarded. Codes that pass the reusability are added up to the repository otherwise they are sent for refactoring.

The refactoring (Ref. req.) process help to avoid the situation where by two similar classes with the same functionality exist in the repository. It is a technique of changing the internal structure of a program while preserving the program behavior [21, 22].

Our reusability test is hinged around the concept proposed by Albaradeen et al. in [20]. There are two test involved: general and specific. General test is required to determine whether a code can be stored in library for later reuse. There are four main attributes associated to this test: i. Co-existence the ability of the code to work on difference platforms (machine independence) ii. Adaptability the degree of which classes can communicate with other classes within a system, in other words the degree to which software classes are independent of one another, the change in one class may not necessarily affect other classes’ iii. Communication communality concerns about the data structure used in the program iv. Software generality concern about the ethics follows while building reusable software. For the purpose of this work the last reusability is not considered since the framework is targeted for the novice programmers. The second reusability test (specific) has three attributes: i. code suitability which determined whether a code performs its intended functions properly ii Documentation to simplify the understanding of the source code and iii Component modifiability to measure how code can be change to meet the requirement of a new system.
D. Repository

For effective components reuse component repository are built to store new components for future reuse. Repository is a structured storage that store, search and retrieve components required for reuse [23]. Components repository are voluminous in nature, housing numerous number of components. The voluminous nature of repository causes considerable challenges of searching and retrieving required components especially when components are accompanied by their documentations. As the repository grow in size, considerable amount of components may have the same or similar functionality, as such storing similar components will only increase the components retrieval time. Other components in repository may not be reusable or their reusability is poor due non frequent access by re-users, therefore such components are required to remove in order to free the repository to save new components as well to decrease access time.

Minton [24] proposed frequency deletion criteria of components stored in repository based on the frequency of component access. Components are set to an initial value (counter) as soon as they are stored in to repository, each time a component is accessed the counter is incremented. Maximum number of components to be stored in the repository is established (overflow limit), if the repository reaches the overflow limit, components with less retrieval frequency are considered for deletion. When repository space becomes free new components can be added except if similar exist. If there are two or more components with the same frequency the oldest is considered for deletion.

In CExtractor, if the frequency deletion criteria are not enough, then the usability test will be used to suggest code for deletion. The usability test is based on the user’s response if the code performs the intended functionality properly, users are asked to rate a code after used. The codes with lowest rate are suggested for deletion. However, the final decision is always up to the CExtractor administrator.

REFERENCES

http://dx.doi.org/10.1145/130844.130856

http://dx.doi.org/10.1109/TSE.2004.50

http://dx.doi.org/10.1145/43857.43859


http://dx.doi.org/10.1145/199688.199770


