

Eclipse Code Monitoring Utility

Aditi Billore¹, Minita Chaudhari², Sneha Desai³, Aishwarya Gote⁴

¹Student, Dept. of Computer Engineering, PCCoE, Maharashtra, India

²Student, Dept. of Computer Engineering, PCCoE, Maharashtra, India

³Student, Dept. of Computer Engineering, PCCoE, Maharashtra, India

⁴Student, Dept. of Computer Engineering, PCCoE, Maharashtra, India

Abstract - Programming exercises for students in the lab sessions require a considerable time and human resources. Tutor have to give separate attention towards each pupil, which is impractical. This paper describes an Eclipse IDE plugin named Eclipse Code Monitoring Utility (ECMU) that will help college professors to assess the performance of each student during practical sessions. Students have to practice their assignments during lab sessions and corresponding professor would be able to assess their proceedings. This plugin will dynamically monitor each student's code and generate a report which will be stored at the repository along with the source codes. This project can be used to enhance students' quality of understanding towards programming. It can be implemented in colleges and universities to monitor the students' carrying out.

Key Words : Eclipse, Programming, Coding, Behaviour, Report, Repository.

1. INTRODUCTION

In present scenario, students do not perform well in practical exams as they do not have logical and practical knowledge. This is because students do not implement the academic assignments by themselves during practical sessions. Moreover, professors have to analyze the involvement and understanding of students. In this paper, the authors are developing a server-client assistance system for instructor to collect, store, monitor and analyze students' codes and programming activity logs [2].

We are developing an eclipse plugin to monitor students' execution during labs. This will be done by dynamically developing a activity log of students' programming activities and making it available to the tutor. Every students' source code and activity log will be stored at server repository. Report for the particular session will then be generated consisting of each student's record of programming which will then be sent to the professor.

1.1 1.1 Eclipse IDE

Eclipse is an Integrated Development Environment (IDE) for different computer programmings like Java, C, C++, Python, Perl, Ada, ABAP, COBOL, D, Fortran, Haskell, JavaScript. It is open source software containing a base workspace and an extensible plug-in system for customizing the environment. Its written mostly in Java [1]. Eclipse provides IDEs and platforms for nearly every language and architecture. These platforms deliver the most extensive collection of add-on tools available for software developers [1].

1.2 Eclipse software component - Plug-ins

The Plug-in Development Environment (PDE) provides tools to create, develop, test, debug, build and deploy Eclipse plug-ins, fragments, features, update sites and RCP products [1].

PDE also provides comprehensive OSGi tooling, which makes it an ideal environment for component programming, not just Eclipse plug-in development [1].

The PDE UI component provides a comprehensive set of tools to create, develop, test, debug and deploy Eclipse plug-ins, fragments, features, update sites and RCP products [1].

2. MATERIALS AND METHODS

ECMU consists of three modules, login in, processing the source code for generating report and repository backup. In order to examine students' interactions in the IDE, we extract all XML log files in ECMU [3].

2.1 System Architecture

System starts with a login window, wherein each student will login with their respective credentials. Then the student has to practice their programming assignments and submit it. Now at the back end, system creates a log file which is continuously updated until the student submits the code.

Further the submitted code along with its log file is stored at the server side repository. Professors will have a web portal wherein they can view the activity of each student for assessment.

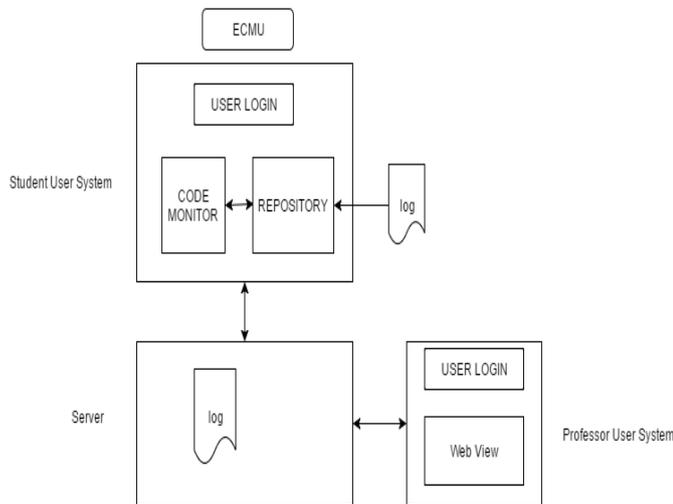


Fig -1: System Architecture

2.2 Client-Server Connection

Students behave as clients and their programs are backed up at the server repository. Along with the source code respective log files developed are also stored at the server side. Server repository will be the central location where a database containing students' information and their programs performed will be stored. Thus, student can access his programs from any host in the college labs.

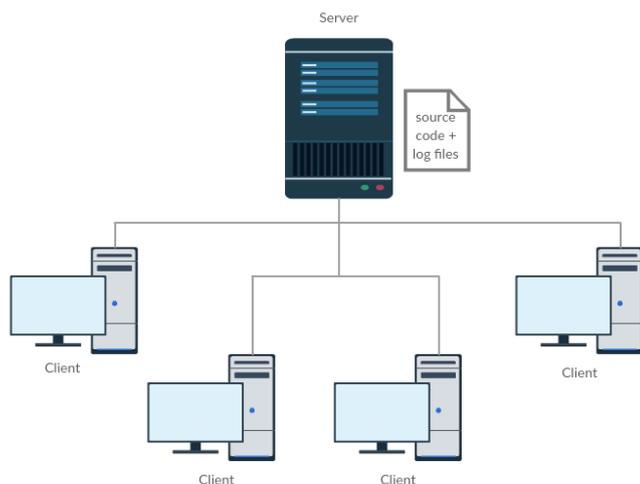


Fig -2: Client-Server Connection

2.3 Code Monitoring

Student's programming will be monitored dynamically using timestamps. For every line in program, a record will be generated and updated to the log file. Line would be detected using delimiters. This log file updation will be done continuously until the program is submitted. Records with equal timestamps will be highlighted. This will inform professor every student's actual involvement in programming, restricting them from copying.

Timestamp	Line
02 December 2016, 14:55:12	Int main()
02 December 2016, 14:55:20	{
02 December 2016, 14:55:23	System.out.println("Hello World");
..	..
..	..

Table -1: Example of records in log file

2.4 Web View for Professors



Fig -2: Web portal for professors

Professor will receive a report from the server system. On basis of it, he can analyze students' involvement, performance and understanding during the lab sessions. If not convinced with any student's involvement he can then

access the student's program directly from the server using the web portal provided by the ECMU.

3. CONCLUSIONS

In this paper, we are proposing an Eclipse plug-in that offers an environment to assess student's programming performance and their involvement and helps professors to determine whether learning goals were reached.

Use of ECMU will lead students to actually practice the assignments that would improve their understanding towards the programming. Also, will help professors to modify their teaching technique if required.

Additionally, the effort required for gathering results by far exceeds the effort for a traditional evaluation and assessment such as an exam, because computer labs need to be organized, the software needs to be deployed, sizable human resources to supervise the students and results need to be gathered directly after the exam. We plan to ease the process by deploying a single system.

REFERENCES

- [1] <https://eclipse.org/>
- [2] Baso Habibi, "Student's programming activity tracking system to help instructors of the programming exercise", Region 10 Symposium (TENSYMP), 2016 IEEE.
- [3] Markus Fuchs, "Monitoring Students' Mobile App Coding Behavior", Global Engineering Education Conference (EDUCON), 2014 IEEE.