

Educational Programming Systems for Learning at Scale

Qianxiang Wang

Institute of Software, School
of EECS, Peking University
Key Lab of HCST (Peking
University), MoE, China
wqx@pku.edu.cn

Wenxin Li

Department of Computer
Science, Peking University
China
lwx@pku.edu.cn

Tao Xie

University of Illinois at
Urbana-Champaign
Urbana, IL, USA
taoxie@illinois.edu

ABSTRACT

Learning programming at scale underlies computer science education ranging from basic programming to advanced software engineering topics. There are strong needs of providing effective system supports for learning programming at scale. Among various desirable characteristics of such system supports, system supports shall allow students to write programs via an online Integrated Development Environment (IDE), allow students to get feedback on how they perform on the given programming exercises, etc. To aim for such effective system supports for learning programming at scale, research teams from Peking University have developed two systems: POP (denoting Peking University Online Programming System) and POJ (denoting Peking University Online Judge System). These two systems have achieved high impact among students around the world (especially those in China). In this paper, we present the overview of the two systems, along with our ongoing and future work on extending the systems for achieving higher effectiveness in supporting learning programming at scale.

POP: PEKING UNIVERSITY ONLINE PROGRAMMING

POP (denoting Peking University Online Programming System, formerly known as CEclipse [2, 3]) has been developed by the Software Engineering Institute of Peking University (its development efforts have been led by the first author since 2009). Figure 1 shows the architecture of POP. POP has initially supported Java development, and recently been extended to support C/C++ along with HTML/PHP (<http://webassist3.seforge.org/phponlineide/>). POP provides the same basic features as the local Eclipse IDE, such as Project Operations, Package Operations, and Class Operations for Java programs, and Compile, Run, etc. POP also provides some advanced features of the local Eclipse IDE, such as Code Auto-Completion, Code Selection, Code Tips (e.g., from Javadoc), and Debugging. In addition, POP supports capture/replay of user behaviors in the online IDE.

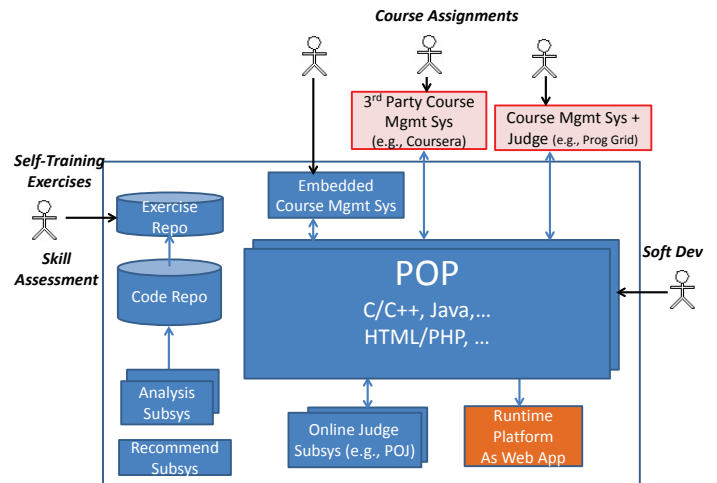


Figure 1. The architecture of POP

There are three main types of users of POP: students in a course for conducting course assignment (as shown near the top of Figure 1), students who do exercises for self-training (as shown on the left side of Figure 1), students (or even practitioners) who leverage POP as an online IDE for software development (as shown on the right side of Figure 1). In the setting of course assignment, POP provides the instructor with an option of configuring POP to prevent copying/pasting of code segments into the online IDE. Such configuration is to alleviate situations where (1) students use their local IDE to write programs and then in the end copy and paste the completed programs to POP, or (2) students copy and paste other students' programs to POP (committing plagiarism).

POP has its own embedded course management subsystem, and has also integrated third-party course management systems such as the one supported by Coursera. POP also supports another subsystem called Programming Grid (<http://programming.grids.cn/programming/>), which is a course management and assignment judge system developed by the Institute of Network Computing and Information Systems of Peking University. POP has its analysis subsystem for analyzing student-submission programs in the code repository. POP also has a recommendation subsystem (currently under development) for recommending exercises to students and supports online judge systems such as POJ (described in the next section).

At Peking University, POP has been used in courses of *Compiler Practice* and *Advanced Software Engineering* since

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).

L@S 2014, March 4–5, 2014, Atlanta, Georgia, USA.
ACM 978-1-4503-2669-8/14/03.
<http://dx.doi.org/10.1145/2556325.2567868>

2010; in 2013-2014, POP is currently used in courses of *Java Programming Design* and *Java Advanced Techniques* along with *Introduction to Computing* (for non-CS majors). For the 2013-2014 course of *Introduction to Computing* (for non-CS majors) at Peking University, there have been more than 1000 homework-assignment submissions using POP. POP has also been used by students at Nankai University and North China University of Technology in China. Since 30 September 2013, a Massive Open Online Course (MOOC) on *Introduction to Computing* offered via Coursera (<https://www.coursera.org/course/pkuic>) has used POP as the course-assignment platform. POP was also used in four programming contests in China.

POJ: PEKING UNIVERSITY ONLINE JUDGE

POJ (denoting Peking University Online Judge System [1] <http://poj.org/>) has been developed by the Artificial Intelligence Lab of Peking University (its development efforts have been led by the second author since 2003). In particular, POJ can compile and test a program submitted for a specific problem against a set of test cases prepared for the problem, and then respond whether the program passes the testing or not. POJ was initially developed to be an ACM International Collegiate Programming Contest (ACM/ICPC) training platform for Peking University students. POJ has been evolved to a general educational platform for both contest training and programming practicing, being used by a large number of students around the world. Later an open platform called Open Judge (<http://openjudge.cn>) has been released to the public for allowing people to construct their online judge systems for programming-related courses. More recently, an online code-evaluation system and algorithm-training platform (<http://codevs.cn/>) has been made available for incorporating an online IDE, categorization of exercises, social networking, etc. From 2003 till recently, POJ has reached 0.2 million users and 8 million submissions.

RELATED WORK

Online IDEs have been gaining popularity in both industry and education. Some example online IDEs are Cloud9 (<https://c9.io/>), CodeAcademy (<http://www.codecademy.com/>), Compiler (<https://compilr.com/>), and Codemoo (<http://www.codemoo.com/>). Compared with these online IDEs, POP provides a richer-feature IDE, which provides the same look-and-feel along with various advanced features of the Eclipse IDE. POP allows users to easily switch between their familiar Eclipse IDE and the POP online IDE.

Pex4Fun [4, 5] (<http://pex4fun.com/>) is a web-based serious gaming environment (released by Microsoft Research) for teaching and learning computer science at scale. Pex4Fun allows students to write programs in a browser with Intellisense support. In its Coding Duel game, students write code to implement a teacher's specification (i.e., a secret sample solution to a given problem), and then Pex4Fun leverages a test-generation engine to find and report any discrepancies in behavior between the student's code and the specification.

POP provides a richer-feature IDE than Pex4Fun's code editor. POJ relies on a given set of test cases to judge the correctness of a student's submission whereas Pex4Fun relies on the underlying test-generation engine and the given specification.

ONGOING AND FUTURE WORK

First, we plan to extend POP to support more programming languages. For example, due to high popularity, Python and Ruby are languages that we plan to support in POP. Second, in both POP and POJ, we are exploring finer-grained grading of student submissions along with giving richer feedback to students when their submissions are not correct yet. Third, we plan to explore research on educational analytics and data mining on user data collected by our systems. POP is able to collect sequences of small edits entered by students when typing in their programs. Both POP and POJ have accumulated a lot of submissions from students around the world along with a lot of exercises created by teachers around the world. We have started some preliminary work [1] on personalized exercise recommendation, automatic contest generator, etc.

ACKNOWLEDGMENTS

Qianxiang Wang's work is sponsored by the National Natural Science Foundation of China (Grant No. 61033006) and the HighTech Research and Development Program of China (Grant No. 2013AA01A213). Tao Xie's work is supported in part by NSF grants CCF-0845272, CCF-0915400, CNS-0958235, CNS-1160603, CNS-1318419, CCF-1349666, and NSF of China No. 61228203.

REFERENCES

1. Lin, S., Zhang, Q., and Li, W. A programmer self-training system with programming skill evaluation and personalized task recommendation. In *Proc. International Conference on E-learning, E-business, Enterprise Information Systems, and E-Government (EEE)* (2013), 112–117.
2. Ling, W., Liang, G., Kui, S., and Wang, Q. CEclipse: An online IDE for programming in the cloud. In *Proc. IEEE World Congress on Services (SERVICES)* (2011), 45–52.
3. Ling, W., Liang, G., and Wang, Q. Program behavior analysis and control for online IDE. In *Proc. Computer Software and Applications Conference Workshops (COMPSACW)* (2012), 182–187.
4. Tillmann, N., Halleux, J. D., Xie, T., Gulwani, S., and Bishop, J. Teaching and learning programming and software engineering via interactive gaming. In *Proc. International Conference on Software Engineering (ICSE), Software Engineering Education (SEE)* (2013), 1117–1126.
5. Xie, T., Tillmann, N., and de Halleux, J. Educational software engineering: Where software engineering, education, and gaming meet. In *Proc. International Workshop on Games and Software Engineering (GAS)* (2013), 36–39.