Elice: An online CS Education Platform to Understand How Students Learn Programming

Suin Kim
School of Computing
KAIST
Daejeon, South Korea
suin.kim@kaist.ac.kr

Jungkook Park
School of Computing
KAIST
Daejeon, South Korea
pjknkda@kaist.ac.kr

Jae Won Kim
School of Computing
KAIST
Daejeon, South Korea
jaewonk@kaist.ac.kr

Alice Oh
School of Computing
KAIST
Daejeon, South Korea
alice.oh@kaist.edu

Abstract

We present **Elice**[1], an online CS (computer science) education platform, and **Elivate**, a system for taking student learning data from Elice and infers their progress through an educational taxonomy tailored for programming education. Elice captures detailed student learning activities, such as the intermediate revisions of code as students make progress toward completing their programming exercises. With those data, Elivate recognizes each student's progression through an education taxonomy which organizes intermediate stages of learning such that the taxonomy can be used to evaluate student progress as well as to design and improve course materials and structure. With more than 240,000 intermediate source codes generated by 1,000 students, we demonstrate the practicality of the Elice and Elivate. We present case studies that confirm that categorizing student actions into the different steps of the taxonomy results in better understanding of the effect of TA's assist and student's performance.

Author Keywords

Online education; online programming; social learning; collaborative learning; computer science education

[1]https://www.elice.io

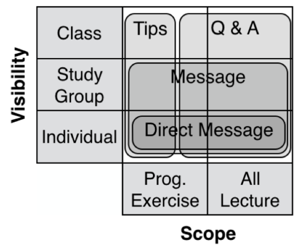

Figure 1: Three channels of sharing the knowledge.

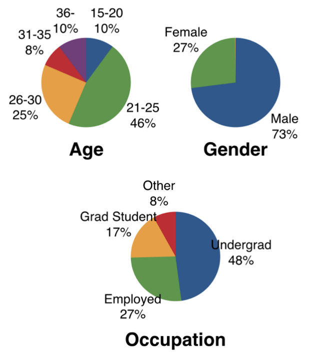

Figure 2: Demographics of the students enrolled in Elice for the machine learning online summer camp.

Introduction

With well-designed technologies, the domain of CS education is quite appropriate for systematic development and research about understanding student progress and assisting learning for individual students in large offline or online classes. In this paper, we present Elice, a Web-based platform, which integrates the three important aspects of analyzing and supporting CS education: 1) collecting and analyzing all actions of the students in doing the programming exercises, 2) mapping students' intermediate steps to an educational taxonomy to understand their progress, and 3) enabling students and TAs to interact using chat, forum, and tips features. We connect the automatic analysis of student progress and online tutoring with a learning taxonomy [2]. For each programming exercise, we map the students' actions to each of the steps in the taxonomy, such that they can be analyzed to infer the students' learning progress in more detail than just completion of the exercise. This allows TAs to help the students to move up (i.e., Elivate) through the learning taxonomy to achieve the maximum attainment.

Elice: An Online CS Education Platform

Elice is an online social CS education platform that supports online programming and collaborative learning. In Elice, students watch lecture videos and work on programming exercises in a Web browser. Elice enables collaborative and social learning with peers, teaching assistants (TAs), and lectures through multiple feedback channels.

Online Programming

We designed online programming to maximize the actual programming experience for students. Elice lets students skip the development environment setups by using a pre-defined container. Each student's code submission is compiled in the pre-defined development container, and the output is displayed back in the student's browser. During this compilation process, all of the student's data including the input data, the submitted code and the output are stored in Elice. When students submit their code for grading, a test script runs and shows grades to students in real-time. Elice allows students to submit their code multiple times for grading and even after receiving full credit for the programming exercises.

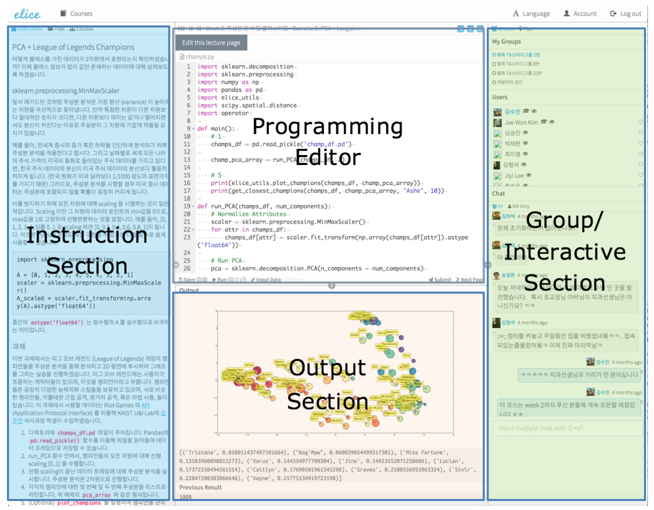

Figure 3: Elice Web-based programming platform. Current screen shows the exercise and result of the PCA algorithm.

Collaborative Learning

We have designed three channels of social interaction that facilitate collaborative learning among peers, TAs and lecturers (Figure 1). These channels support varying time spans of collaboration, from instant to delayed, and have

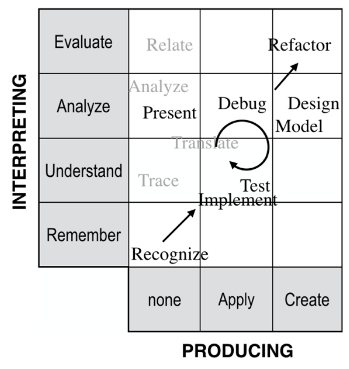

Figure 4: Fuller's CS-specific educational taxonomy.

Step	Meaning
Recognize	Read a lecture material
Design & Model	Devise a solution structure
Imple-ment	Translate the solution to code
Test & Debug	Apply the set of test cases;detect and correct flaws
Refactor	Try to get a better answer after achieving full credit
Present	Share tips and provide answers

Table 1: 6 steps of learning process for programming exercises in Elice.

different scope and visibility. In Elice, a class includes all students, TAs and lecturers who are registered in the same online course. Students in the same class may be assigned to a group consisting of two or more students.

Machine Learning Online Summer Camp
In August 2015, we recruited students for an online summer camp to learn about basic machine learning algorithms. The initial registration was about 4,000 students, and we selected around 1,000 of them. Then, we randomly assigned these students to 30 TA Groups with 10 students each, 31 No-TA Groups with 10, 20, or 30 students each, and the rest worked individually (No-Group). TA-Group had ten students each and one TA who logged in two assigned days in a week for 1.5 hours to guide the students during these online office hours. Students in No-TA-Group had no assigned TA, and No-Group took the course individually and had no access to the chats. Demographics of the students are shown in Figure 2. Each week included short introductory lecture videos for machine learning algorithms and eight to ten programming exercises, in total 35 programming exercises for four weeks.

Elivate: Identifying the Learning Process

In education research, a widely-cited educational taxonomy by Bloom [1] organizes the "higher order thinking skills" from simple to complex. This taxonomy can be used by a lecturer to define the objectives of a course, as well as evaluate the learning stages of a learner. In this paper, we adopt Fuller et al.'s definition of educational taxonomy [2], specifically tailored for computer science programming education (Figure 4). Fuller's taxonomy was developed based on the findings that there are semi-independent abilities in learning to program, ability to understand and to produce. Using 12 problem-solving activities from [2], we re-define six steps of learning process, shown in Table 1.

Mapping student activities to programming education taxonomy
We map between student activities that can be detected and logged using Elice and the stages in the two-dimensional programming education taxonomy. Recognize can be easily detected by checking if the user reads the lecture material. Implement and Test & Debug correspond to each code submission, and we detect using student's code and input data. Refactor is the most advanced stage, and any code submission after getting a 100% score falls into this category. Present involves the activity of sharing, which can be done through three channels: chat, tips, and board.

Case Studies

Using the data collected from Elice, we analyze two important questions from the perspective of educational research.

Case 1: Overview – The 3 levels of TA assistance
Our study incorporates three levels of feedback for students in the three groupings: TA-Group, No-TA-Group, and No-Group. When we look at the distribution of the stages in the educational taxonomy (Figure 5), the difference is quite clear. Average students in TA-Group made significantly more code submissions compared to No-TA-Group and No-Group students ($p < 0.001$). Also, it is notable that Refactor, the most advanced stage in the computer science education taxonomy, is the highest achieved from students in TA-Group. This implies that higher levels of assistance help student to "traverse the learning process" to maximum attainment.

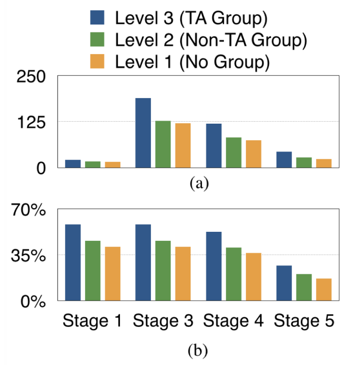

Figure 5: Code submissions from students receiving different levels of assistance. (a) Count of learning activities. (b) Average percentage of users achieved the learning stage.

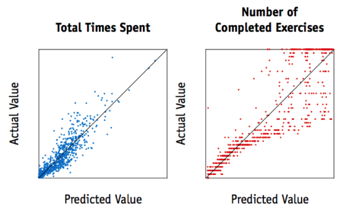

Figure 6: Prediction of the two final performance statistics using student's learning progress from the first two weeks. Pearson's r values are 0.914 an 0.905, respectively.

Case 2: Iterative transitions in the learning process
Starting from the Recognize process, by reading a lecture material, a student internally Design & Models the structure of the solution. After then, during the Implementation stage the student translates the design into source code. Test reveals the error of the code, so the student Debugs by patching the design structure of the solution. Then Implementation and Test & Debug occur iteratively. After analyzing submissions from 214 students who completed the course, we found that students in TA-Group, on average, significantly have higher number of iterations while solving the programming exercises (Figure 5).

Case 3: Predicting student's learning performance
Predicting whether a student will complete the class including all exercises is important for online classes. An early prediction may enable early personalized feedback to motivate and assist students who may be having difficulties. In this case study, we predict two variables: (i) total time spent and (ii) number of completed exercises. We trained a simple linear regression with OLS to predict two variables using the features of student's learning process at first two weeks. With Pearson's r value for total times spent and number of solved exercises are 0.914 and 0.915, additional correlation analysis verifies that the proposed taxonomy-related features show higher correlation score compared to conventional features (Figure 6).

Discussion and Future Work

The number of students taking CS courses is rapidly increasing. In order to support ever-growing number of students, it is essential to provide online programming environment and find the status of students for appropriate guidance and feedback. Our online social CS education platform, Elice, allows students to start programming immediately in their Web browser and collaborate with peers and TAs when they struggle. All of these student's data, including interactive activities to intermediate code submissions, are collected in Elice. From the collected data, we have identified the effect of assistance and the six steps of learning process in CS education. Such findings confirm that the important role of TAs when it comes to learning experience even in online setting. Therefore, in order to support the large number of students with limited number of TAs, it is vital to increase the efficiency of TAs. The six steps of learning process defined in our study help TAs identify the status of students quickly and increase the productivity of TAs.

We need to explore better means of helping students to move on to the next stage in the learning taxonomy. One possible method could be pushing appropriate tips or hints to students if the system finds that students are struggling or spending too much time in each stage.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.10041313, UX-oriented Mobile SW Platform).

References

[1] Bloom, B. S., and Engelhart, M. D. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. David McKay, 1969.

[2] Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., Lahtinen, E., Lewis, T. L., Thompson, D. M., Riedesel, C., et al. Developing a computer science-specific learning taxonomy. In *ACM SIGCSE Bulletin*, ACM (2007).