Exploring the Complexity of Crowdsourced Programming Assignments

Nea Pirttinen
University of Helsinki
Helsinki, Finland
nea.pirttinen@helsinki.fi

Juho Leinonen
Aalto University
Espoo, Finland
juho.2.leinonen@aalto.fi

ABSTRACT
CrowdSorcerer is a tool in which students can create their own programming assignments according to teacher’s instructions, and later review their peers’ assignments. In this lightning paper, we take a brief look into the level of complexity of assignments novice programmers create with the tool.

Author Keywords
crowdsourcing, learnersourcing, assignment creation

INTRODUCTION
Crowdsourcing has been used in computer science education mostly for student-based assignment creation. For example, PeerWise [1] allows students to create multiple-choice questions and revise course topics using questions created by their peers. In a similar way, computer science education tools like CodeWrite [3] and CrowdSorcerer [7] assist students in creating their own programming assignments according to guidelines given by the instructor.

Education tools like these allow students to approach the course topics from a different perspective than usual. If the tool also incorporates peer reviewing, which most of the tools do [1, 7] in some way, the students can also revise the course concepts through the review process. At the same time, collecting small assignments, programming-related or not, can help teachers to build a database of suitable exercises for quizzes or drill practice.

Peer review is widely used and often accurate enough in terms of coverage or quality [2, 4, 8, 6], at least for cases where the reviews will not directly affect the students’ grades. However, since students are most often novices in the course topics, it is important to ensure that the quality of the collected assignments is reasonable, even after peer review. In this paper, we take a brief look into the complexity of programming assignments created by students during the second week of an introductory programming course.

Our research question is What types of programming assignments do novice students create? To answer this, a set of 91 assignments was categorized according to their types and features. The categorization identified seven different categories based on the complexity of the assignments.

CROWDSORCERER

Figure 1. The basic assignment creation view of CrowdSorcerer. The assignment field contains a student-written assignment handout. The source code field has a student-written code with the model solution marked with the checkboxes on the left (blue lines). The lines in gray form a template that cannot be edited by students.

The data was collected using CrowdSorcerer [7], an embeddable tool used for programming assignment creation. The source code field has a student-written code with the model solution marked with the checkboxes on the left (blue lines). The lines in gray form a template that cannot be edited by students.

Our research question is What types of programming assignments do novice students create? To answer this, a set of 91 assignments was categorized according to their types and features. The categorization identified seven different categories based on the complexity of the assignments.

The data was collected using CrowdSorcerer [7], an embeddable tool used for programming assignment creation. The source code field has a student-written code with the model solution marked with the checkboxes on the left (blue lines). The lines in gray form a template that cannot be edited by students.

The user interface of CrowdSorcerer is built with React 1. The backend 2 is a Ruby on Rails application and holds most of the system’s functionality and storage. The student-created assignments are sent to a Test My Code programming assignment evaluator server [9] 3 to assess whether the assignments pass the student-created test cases.

The user interface for basic assignment creation can be seen in Figure 1. First, the user writes an assignment handout according to specifics given by the instructor. These instructions may

1https://github.com/rage/crowdsorcerer
2https://github.com/rage/crowdsorceress
3https://github.com/testmycode
The categorized assignments are found in Table 1. The student-created programming assignments were categorized according to their features. Since the instructions prompted for an assignment that asks the user for an integer input, uses a conditional statement and prints a string output, the simplest expected assignment should have at least one if-statement. For tests, give an example input and the output the program will print with this input.

The data was collected from 91 students on an introductory Java programming course who gave their permission for using their data for scientific purposes, and completed an assignment using CrowdSorcerer. The instruction for the CrowdSorcerer assignment was: “Create an assignment that requires a student to create a program that reads an integer from the user, uses a conditional statement to inspect the integer and then prints a string. For tests, give an example input and the output the program will print with this input.”

The following assignment is an example of a typical, relatively simple program, categorized as intermediate if-else. The assignment uses multiple if-statements one after another, but not nested or if-elses.

```java
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.println("How many cats do you have? ");  
    int cats = Integer.valueOf(sc.nextLine());  
    if (cats == 0) {  
        System.out.println("What a pity!");  
    } else if (cats >= 1 && cats <= 4) {  
        System.out.println("Exemplary work!");  
    } else if (cats > 4) {  
        System.out.println("Wow, what a clowder!");  
    }
}
```

Altogether, these preliminary results indicate that students mostly follow instructions given when creating crowdsourced programming assignments. The quality of these assignments could be evaluated, for example, through peer review, and after enough assignments have been collected, the best ones can be used to build a database of small programming assignments. These can be reused in future courses, not only in our context, but shared with the computer science education community. In our future work, we are interested in studying assignment quality in more detail, and exploring whether students from different demographics create different kinds of assignments.
REFERENCES


