Personalized Feedback for Open-Ended Novice Programming

Michelle Ichinco
Department of Computer Science
and Engineering
Washington University in St. Louis
St. Louis, MO, USA
michelle.ichinco@wustl.edu

Caitlin Kelleher
Department of Computer Science
and Engineering
Washington University in St. Louis
St. Louis, MO, USA
ckelleher@wustl.edu

Author Keywords

Novice programming; online education.

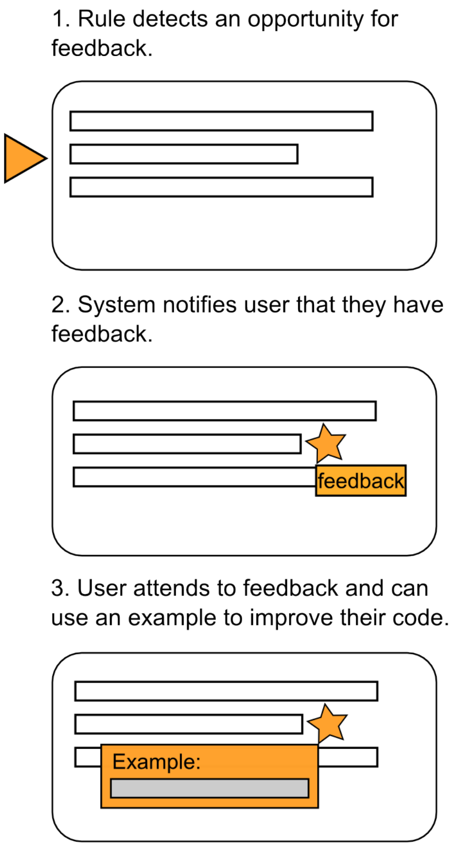
ACM Classification Keywords

H.5.2. [User Interfaces]

Introduction

Although the number of programming jobs is growing faster than the number of workers trained in programming, computer science education in schools is often still lacking [3]. In place of formal education, novice programmers can use online resources to learn programming, such as MOOCs, tutorials, games, and programming environments. These systems and resources fall into two groups: either 1) users complete predefined tasks with correct and incorrect answers, which the system can check, or 2) users work on their own projects, but only receive feedback for errors. Since open-ended projects can be highly motivating for students, systems must be able to provide feedback that can help these programmers improve their programming skills. One way to do this is to provide feedback similar to the feedback teachers provide to students in a classroom: ideas for ways a student could improve their work and learn along the way.

We propose large-scale support for learning programming independently: providing unsolicited feedback to novice programmers in the form of examples. We propose a way to provide feedback to novice programmers working on open-ended projects. The feedback is distributed using "rules" that check whether a program would benefit from a certain suggestion. The feedback utilizes example code, which is widely available online and can be effective as a learning resource.


1. Rule detects an opportunity for feedback.

2. System notifies user that they have feedback.

3. User attends to feedback and can use an example to improve their code.

Example Feedback System

There are three main components of this system: rules, suggestions, and examples.

Rules

Rules are scripts that determine whether a program should receive a certain suggestion. A rule can focus on code quality issues, opportunities to introduce undiscovered system features, or opportunities to introduce new programming concepts [2].

Suggestions

When first presented, a suggestion only hints at the content by providing a short description of what the programmer can learn. Upon exploring the suggestion further, the programmer can view a code example designed to help the programmer learn a new concept while improving their program. The point of this design is to enable the user to understand the content of the feedback so that they can decide whether it is relevant before delving into the details of how to actually use the example. We hypothesize that this should make the feedback less overwhelming, because a learner can start to understand the idea of the suggestion before even looking at example code.

Examples

Examples have been widely and effectively used as educational material, especially in the form of worked examples [4]. Additionally, for programming, many examples are available online, making them easy to acquire. However, example code snippets on their own are often difficult for novices to use [1]. This type of system will need to provide support for example use in order for novices to be able to use and learn from code examples.

Income Generation and Online Education

While our work has not previously addressed the idea of income generation for students, the proposed feedback system could help students transition from a purely educational situation to a paid opportunity in which they continue to receive personalized feedback about their work. Though our work focuses on programming, this idea may also apply to other educational areas, such as foreign languages. In general, this type of system could work for any topic where similar rules could be created to check for learning opportunities and examples can be found online.

References

1. Michelle Ichinco and Caitlin Kelleher. 2015. Exploring novice programmer example use. *VL/HCC, 2015,* 63–71.
2. Michelle Ichinco, Aaron Zemach, and Caitlin Kelleher. 2013. Towards generalizing expert programmers' suggestions for novice programmers. *VL/HCC, 2013*, 143–150.
3. National Center for Women & Information Technology. 2011. *Computing Education and Future Jobs: A Look at National, State & Congressional District Data*.
4. John Sweller and Graham A. Cooper. 1985. The Use of Worked Examples as a Substitute for Problem Solving in Learning Algebra. *Cognition and Instruction* 2, 1: 59–89.