Scaffolding, the zone of proximal development, and novice programmers

Nadia Kasto Awbi*, Jacqueline Whalley, and Anne Philpott

School of Computer and Mathematical Sciences, Auckland University of Technology, New Zealand

Abstract

The work, which is part of a doctoral research project, reported here aims to explore the learning strategies that novice computer programmers adopt when writing code, the ways in which they integrate knowledge, and the processes they employ when applying their knowledge and skills in different contexts. Here we present an analysis of the data obtained using think-aloud retrospective interviews of two novice programmers attempting to solve a set of programming tasks. Our findings, based on a narrative analysis of these interviews, indicate that scaffolding can influence progression in learning and can extend a student’s zone of proximal development [5].

Introduction

In recent years, researchers have focused on the Bloom and SOLO taxonomies [1] and Neo-Piagetian levels of development [3] as possible sources of explanation of students’ abilities to reason about code. A recent study into the cognitive aspects of the early stages of learning to write computer programs suggested that with the right behavioural approaches to learning students are able to expand their zone of proximal development (ZPD) [4].

• Research Questions

In this work, which is an extension of the work reported in [5], we were interested in discovering:
• What kind of tasks scaffold and reinforce code writing?
• Can we identify the Zone of Proximal Development (ZPD) of a student? Does it change?

Method

One-on-one think aloud retrospective interviews were conducted with AUT students from an introductory Java programming course. These sessions were held once a week for 10 weeks, were recorded using a video camera and typically lasted about 60 minutes. Each session students were asked to answer short code writing questions. In this paper we report on and analysis of the interviews from two students who were both in the top quartile of the class. If a student was unable to answer a question unaided the interviewer then provided assistance as soft scaffolds where the construction of the scaffold is occurs as instruction is occurring.

In order to analyse the results we classified the level of assistance the interviewer was further classified as soft scaffolding using Perkins and Martin’s system as either clarify, general prompts, hint, or exact solution [2]. The ZPD can be defined as “the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under guidance ” ([4] p. 86). We therefore identified that a student was within their ZPD if they could solve a problem with scaffolding of the clarify, general prompts or hint types. A student was considered to be within their comfort zone (CZ) if they were able to solve a given problem independently and outside of their ZPD if they were unable to solve the problem.

Results and Discussion

The four questions, discussed in the poster, were designed using a robot world and each question provided a small incremental increase in the conceptual complexity of the task. The students did not progress to the next question until they were able to solve the previous question. The students were asked to write a procedure to:

Question1: Write a procedure to calculate the length of a single corridor.

Question2: Write a procedure to compare the length of two corridors and print out a message that either states the corridors were of equal length or gave the length of the longer corridor.

Question3: Write a procedure to calculate the length of the longest corridor.

Question4: Write a procedure to calculate the length of the shortest corridor.

To solve question 2 Andre used two variables to hold the lengths of the two corridors and compare them. However, for question 3 he needed a hint to realize that a most wanted holder variable was required. He also required a second hint in order to update the most wanted holder variable correctly. In Danny’s case (Figure 2, right) questions 1 and 2 were within his comfort zone. Question 3 was clearly outside of his ZPD, but model answer code was discussed with the interviewer in the retrospection phase. Question 4 required the use of the same program schemas but required the length of the shortest rather than the longest corridor to be calculated. Danny was able to recognize the similarity and arrive at a solution to question 4 with minimal intervention in a follow up interview. Therefore in this case it appears that the exact solution to question 3 provided a scaffold that allowed Danny to successfully solve question 4 and also extend his ZPD.

Conclusion

We have demonstrated that it is possible to observe a student’s ZPD and that appropriate scaffolding enables students to extend their ZPD and CZ. We also found that if used appropriately model answers can help a student’s development. We have found that it is possible to learn from a model answer in cases where the model answer allows the students to move forward onto a similar but different question that leads to a new understanding. These findings suggest that Lev Vygotsky’s ZPD theory should be a useful tool for informing teaching practice and formative assessment design in computer programming.

References


