

article:2424**Novices' Knowledge Construction of Difficult Concepts in CS1**

This was a cognitive study that examined how students come to know and apply three basic programming concepts that are known to be difficult: functions, recursions, and arrays [1]. When addressing attrition and dropout issues in CS, this study took a bottom up approach. That is, it was more important to look at how students construct and use their understandings (or misunderstandings) of these abstract concepts. With such knowledge, we can begin building instructional methods and tools to facilitate correct understanding and diffuse incorrect ones.

Clinical interviews and process tracing methods were used in this study as a tool to look into students' conceptual understanding and problem solving processes [2, 3]. Eight undergraduate students were recruited from a 9 week summer session CS1 course in UT Austin's Computer Sciences department. They were presented with pen-and-paper-based problems covering functions, recursion, and arrays. There were two sets of interviews. The first interview occurred during the week that the instructor of the course taught recursion and arrays. The second interview occurred approximately two weeks later with variances on the original tasks. The clinical interviews and process tracing methods allow for probing and extension questions during the problem-solving which gives a fuller cognitive snapshot.

A grounded theory analysis of the data was used to construct a general theory based on the data to explain how students understood these concepts and applied them [4]. The data included interview transcripts, student created artifacts, and demographic surveys.

This study also revealed that students have different types of knowledge bases from which they draw. Moreover, this study illustrated several cognitive processes students undergo in applying their knowledge in programming and problem-solving. The three types of knowledge found were: fragmented, banked (memorized), associated (knowledge that exists due to knowing about something else), and conceptual knowledge (learned knowledge). When working with the difficult concepts presented in the clinical interviews, students' problem solving included an unawareness of detail or problems, the need to code or think of the solution through code, generalizing the solution, and constructing an optimal solution. Having a model of the types of knowledge and the way they are used in computing is an essential tool for instructors when dealing with students of varying ability, especially when dealing with high-level and complex concepts that are abundant in computer science.

References

- [1] N. B. Dale, "Most Difficult Topics in CS1: Results of an Online Survey of Educators," inroads - The SIGCSE Bulletin, vol. 38, pp. 46-53, 2006.
- [2] H. P. Ginsburg and S. Opper, Piaget's Theory of Intellectual Development, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 1987.
- [3] J. R. Hayes and L. S. Flower, "Uncovering Cognitive Processes in Writing: An Introduction

to Protocol Analysis," in Research on Writing: Principles and Methods, P. Mosenthal, L. Tamor, and S. A. Walmsley, Eds. New York, NY: Longman, 1983.

[4] A. Strauss and J. Corbin, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Thousand Oaks, CA: SAGE Publications, Inc., 1998

Author: Timothy Yuen; timster@timothyuen.com

Article

Link:

<http://portal.acm.org/citation.cfm?id=1345375.1345413&coll=GUIDE&dl=GUIDE&CFID=35014869&CFTOKEN=56797477>

[: Back to 2009 Spring/Summer Issue Vol. 4, No. 3](#)

[: Back to List of Issues](#)

[: Back to Table of Contents](#)