The Yale University Cognition And Programming Project

Stephen Slade

Computer Science Department
Yale University

The Cognition and Programming Project (CAPP) in the Computer Science Department at Yale University is an interdisciplinary group exploring a wide range of issues in programming. The current personnel are:

- Dr. Elliot Soloway, Assistant Professor; Dr. Kate Ehrlich, Research Associate
- Lewis Johnson; Jeff Bonar; Valerie Abbott

Work is currently in progress on the following projects:

1. What do experts/novices know about programming? The objective here is to develop a plan-based theory of programming knowledge and to test it empirically with novice and expert programmers. We are also using this theory to study the bugs and misconceptions of novice programmers.

2. Cognitively appropriate programming language constructs. The design of programming languages has largely been determined by computational constraints. However, we have identified cases in which the underlying strategy required by a programming language construct is not the one people normally know or use, and we have shown empirically that they have significant difficulty in learning to use such constructs effectively. We have also shown that when the language construct agrees with people’s natural problem solving strategies they can learn to use such constructs effectively. The implication is that language designers should be more sensitive to cognitive capabilities which people bring to programming — and that computing educators should be aware of the systematic misconceptions which arise due to cognitively poor programming language constructs.

3. Cognitively-based complexity measures of programs. Using our theory of programming plans, we are developing measures of program complexity that are based on the underlying mental effort needed to understand programs. This approach is in contrast to typical measures of program complexity which are sensitive to only surface features of programs.

4. Procedurality/Non-procedurality in problem solving. Algebra word problems are notoriously difficult for students to solve. We have found, however, that students can correctly solve such problems when asked to write computer programs. Moreover, we have recently gathered evidence that the benefits of programming can even transfer, and aid programmers using algebra. The basis of this effect seems to lie in the procedurality inherent in programming, as contrasted with the non-procedurality usually associated with traditional algebra. Our work suggests that if programming languages are made more non-procedural...
then many individuals will not be able to acquire such languages; moreover, it appears beneficial from a problem solving perspective to teach procedural programming languages.

Intelligent Programming Tutor. Complementing the empirical projects, we are actively engaged in building an AI-based tutoring system, PROUST, which can assist novice programmers as they are learning to program. The goal of this system is to identify non-syntactic bugs in a student's program, and then tutor the student with respect to the misconceptions underlying the bugs. This project serves as an excellent test bed for our evolving theory of programming knowledge, and it also serves to raise the next series of questions which a theory of programming must answer.

Recent CAPP publications are listed below.


---

3 This project is currently being funded by the Army Research Institute, under grant number MDA903-80-0068.