Towards More Natural Functional Programming Languages

Invited Talk

Brad A. Myers
Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
bam+@cs.cmu.edu
http://www.cs.cmu.edu/~bam

ABSTRACT

Programming languages are the way for a person to express a mental plan in a way that the computer can understand. Therefore, it is appropriate to consider properties of people when designing new programming languages. In our research, we are investigating how people think about algorithms, and how programming languages can be made easier to learn and more effective for people to use. By taking human-productivity aspects of programming languages seriously, designers can more effectively match programming language features with human capabilities and problem solving methods. Human factors methods can be used to measure the effects, so unsubstantiated claims can be avoided.

This talk will present a quick summary of new and old results in what is known about people and programming, from areas that are sometimes called "empirical studies of programmers" and "psychology of programming." Much is known about what people find difficult, and what syntax and language features are especially tricky and bug-prone. Our new research has discovered how people naturally think about algorithms and data structures, which can help with making programming languages more closely match people's problem solving techniques.

Categories and Subject Descriptors

D.1.7 [**Programming Techniques**]: Visual Programming. D.2.6 [**Software Engineering**]: Programming Environments–*Graphical environments, Interactive environments*. D.3.3 [**Programming Languages**]: Language Constructs and Features. F.3.3 [**Logics and Meanings of Programs**]: Studies of Program Constructs. H.1.2 [**Models and Principles**]: User/Machine Systems–*Software psychology*. H.5.2 [**Information Interfaces and Presentation**]: User Interfaces. I.3.6 [**Computer Graphics**]: Methodology and Techniques–*Languages.*

General Terms

Design, Human Factors, Languages

Keywords

End-user programming, psychology of programming, empirical studies of programming, natural programming.

EXAMPLES

As part of his PhD thesis [1], my student John Pane performed three formative studies to see how non-programmers naturally thought about algorithms for manipulating graphics and numbers. In the first study, we showed 10-year old children pictures of various scenes from the PacMan game, and asked how they would implement them. In the second study, we showed children various situations in a database, and asked them to perform arithmetic operations. In both cases, we had independent analysts evaluate the answers looking for patterns. One of the results was that people consistently operated on sets of objects, rather than iterating or recursing through the set [3]. For example, people said, "When PacMan eats all of the dots, he goes to the next level," and "Subtract 20,000 from all elements in Round 2." Another result is that most people tended to use an event-based style for graphics, such as "If PacMan hits a wall, he stops." In contrast, some people used a constraint style: "PacMan cannot go through a wall."

In extensive study of Boolean expressions, we found that children and adults use words such as "AND," "OR" and "NOT" with inconsistent meanings [2]. For example, "AND" often was used where the Boolean operator "OR" would be required, as in "Scores of 10,000 and up are extraordinary" (since no score can be 10,000 AND up at the same time). Another result is that people were not consistent in the precedence that they expected for operators. For example "Not A or B" often meant "Not (A or B)", but not always.

Using the results of these and prior studies and human-factors principles we have created a new programming language for children called HANDS. User studies showed that novel features of HANDS made it easier for non-programmers to create programs.

REFRENCES

1. Pane, J., *A Programming System for Children that is Designed for Usability.* PhD Thesis, Computer Science Department Carnegie Mellon University, 2002, Pittsburgh, PA. Computer Science Technical Report CMU-CS-02-127.
2. Pane, J.F. and Myers, B.A. "Tabular and Textual Methods for Selecting Objects from a Group," in *Proceedings of VL 2000: IEEE International Symposium on Visual Languages.* 2000. Seattle, WA: IEEE Computer Society. pp. 157-164.
3. Pane, J.F., Ratanamahatana, C.A., and Myers, B.A., "Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems." *International Journal of Human-Computer Studies*, 2001. **54**(2): pp. 237-264.