

Keynote Talk

Inherent vs. Accidental vs. Intentional Difficulties in Programming

Brad A. Myers

Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA USA
bam@cs.cmu.edu
<http://www.cs.cmu.edu/~bam>

Abstract

It is the conventional wisdom that some aspects of programming are difficult to learn, and some aspects are error-prone even for experts. Is it possible to separate what is inherently difficult, and therefore most appropriately dealt with through education, versus what is just accidentally difficult, so a new design for a language or development environment might be able to “fix” the problem? And are there aspects that a designer makes difficult “on purpose”? For example, compare recursion, the syntax for switch statements in C, and how unification works in Prolog, respectively. It is not clear that the conventional wisdom on this topic can be trusted. For example, whereas most argue that concurrency is inherently difficult, the creators of the Alice language argue that they have found a way to make it understandable to novices. This presentation explored some HCI research on this topic, and approaches for identifying the differences.

Categories & Subject Descriptors: D.2.6 [Programming Environments]: Integrated environments; H.1.2 [User/Machine Systems]: Software psychology; D.3.3 [Language Constructs and Features]

General Terms: Documentation, Design, Experimentation, Human Factors, Languages.

Keywords empirical studies of programmers; novice programming; usability of IDEs.

Bio

Brad A. Myers is a Professor in the Human-Computer Interaction Institute in the School of Computer Science at Carnegie Mellon University. He is an ACM Fellow, winner of six best paper awards, and a member of the CHI Academy, an honor bestowed on the principal leaders of the field. He is the principal investigator for the Natural Programming Project and the Pebbles Handheld Computer Project, and previously led the Amulet and Garnet projects. He is the author or editor of over 400 publications, including the books "Creating User Interfaces by Demonstration" and "Languages for Developing User Interfaces," and he has been on the editorial board of five journals. He has been a consultant on user interface design and implementation to over 70 companies, and regularly teaches courses on user interface design and software. Myers received a PhD in computer science at the University of Toronto where he developed the Peridot user interface tool. He received the MS and BSc degrees from the Massachusetts Institute of Technology during which time he was a research intern at Xerox PARC. From 1980 until 1983, he worked at PERQ Systems Corporation. His research interests include user interface development systems, user interfaces, handheld computers, programming environments, programming language design, programming by example, visual programming, interaction techniques, and window management. He is a Senior Member of the IEEE, and also belongs to SIGCHI, ACM, and the IEEE Computer Society.