ARTICLES

10 Abstraction Techniques in Modern Programming Languages

Mary Shaw

Modern programming languages depend on abstraction: they manage complexity by emphasizing what is significant to the user and suppressing what is not.

28 Gambit: A Prototyping Approach to Video Game Design

Tracy Larrabee and Chad Leland Mitchell

Video game designers don't have to be assembly language programmers. Their creative talents can be expressed in a prototyping language.

39 US&R: A New Framework for Redoing

Jeffrey Scott Vitter

Undo, Skip, & Redo—a new interactive approach to user recovery—offers significant advantages over current undo/redo packages.

54 An Interleave Principle for Demonstrating Concurrent Programs

Ted G. Lewis, Keith R. Spitz, and Paul E. McKenney

Although they want to know whether their concurrent programs are working properly, programmers may not need a rigorous proof of correctness.

66 Software Prototyping Using the SETL Programming Language

Philippe Kruchten, Edmond Schonberg, and Jacob Schwartz

With increased computational power available, programming languages can now balance efficiency with expressivity. This flexibility improves rapid prototyping of complex design changes.

77 Generating a Production Compiler from an Attribute Grammar

Rodney Farrow

Must automatically generated compilers use inordinate amounts of memory? Here is evidence that AG-based systems hold the potential for efficient commercial operation.

SPECIAL FEATURE

94 Author, Subject Index—Volume 1, 1984

DEPARTMENTS

7 Letters to the Editor

98 Software Standards: When does more documentation mean less work?

100 Software Reviews: Selecting an Implementation Language—Not Just Between C or Pascal

104 New Products

110 Product Highlights

112 Soft News

118 Call for Papers

119 Classified Ads

120 Professional Calendar

121 Short Courses

122 Book Reviews: Courseware

128 Advertiser/Product Index

Reader Service Cards, p. 128A;
IEEE-CS Membership Application, p. 117.