It seems like an obvious solution to reduce the tremendous investment required to write new software: Why not reuse parts of existing programs and save on development time and testing costs? One reason reusability has been no more than a suggestion is that the tools to make reusability work didn’t exist. But now these tools are appearing, and this issue’s articles show you several of them. With them, reuse may become a reality.

Cover photo by Tony Bullard
Cover design by Kevin Reagan

DEPARTMENTS

5 Software Letters Applying Cocomo • Strategic Defense Initiative
86 Software Standards Preventing Most-Probable Errors in Planning
89 Software Reviews Ada Environments: TeleGen2 • Ada World
92 New Product Reviews BSW-Make • MathPlan • Norton Utilities • SRMS • FTL Modula-2 • MathCAD • Word For Word • Wines on Disk
97 New Products
100 Product Highlights
101 Soft News Steve Jobs Keynotes Unix Conference • NSF Computing Chief on Research Directions • Reusability Libraries
107 Calendar • Call for Papers
108 Book Reviews Programming Productivity • Files and Databases: An Introduction • Handbook of Software Maintenance • Document Formatting and Typesetting on the Unix System • The C++ Programming Language
111 Computer Society Roster
Reusability Comes of Age
Will Tracz
Reusability has long held out the promise of issuing in software's industrial revolution, of transforming a cottage industry into a mass-production system. The tools to do this are now appearing.

Frame-Based Software Engineering
Paul G. Bassett
One of reusability's main problems is how to easily modify available components. This frame-based approach handles the problem.

Melding Software Systems from Reusable Building Blocks
Gail E. Kaiser and David Garlan
This declarative language takes the best features from the three most popular reusability approaches, but overcomes their flaws. It supports language independence, component composition, and tailoring.

The Reusable Software Library
Bruce A. Burton, Rhonda Wienk Aragon, Stephen A. Bailey, Kenneth D. Koehler, and Lauren A. Mayes
The RSL couples a passive database with interactive design tools to make reuse an integral part of the software development process.

Software Reuse through Building Blocks
Manfred Lenz, Hans Albrecht Schmid, and Peter F. Wolf
Specification, design, and code can all be reused easily if handled as a building block. An IBM group recently developed this concept and applied it to systems programming — with success.

Reusability Issues and Ada
Anthony Gargaro and T.L. (Frank) Pappas
How do you write reusable code when your methodology doesn't address reusability? These guidelines developed by a major defense contractor may help.

Can Programmers Reuse Software?
Scott N. Woodfield, David W. Embley, and Del T. Scott
An experiment asked programmers untrained in reuse to evaluate component reusability. They did poorly. Are reusability's promises hollow? Or are there some answers?

Cognitive View of Reuse and Redesign
Gerhard Fischer
Reusable components are not enough. Program designers need tools that help them understand the components and how to use them. Fortunately, some support tools do exist.

Implementing and Optimizing Lisp for the Cray
J. Wayne Anderson, William F. Galway, Robert R. Kessler, Herbert Melenk, and Winfried Neun
This Portable Common Lisp version for the Cray couples the language's strengths with the machine's power. The researchers who developed the implementation tell how they did it.