Componentization: The Visitor Example
pp. 23-30
Bertrand Meyer and Karine Arnout

Patterns, a major advance in software architecture, provide a common vocabulary and a widely known catalog of design solutions that address frequently encountered situations. But they do not support reuse, which assumes components—off-the-shelf modules ready to be integrated into an application through the sole knowledge of an abstract program interface.

To go beyond patterns by turning some of them into reusable components, authors analyze the componentization process through the example of an important pattern, Visitor. They offer a reusable solution that is not only easier to use but also more general than the pattern because it removes its known limitations.

Three Public Enemies: Cut, Copy, and Paste
pp. 31-35
Zoltán Ádám Mann

The problems arising from copied code are not new, and many researchers have investigated how to automatically find copied code segments. An often recommended approach is to refactor the copied code—generalize the segments to a meaningful programming language abstraction, such as function, macro, or class, and replace the segment’s occurrences with references to that abstraction. However, factoring is not always the method of choice and is obviously not a solution for other kinds of text editing, which suffer from the same copy-paste deficiencies.

Given the extensive use of copy-paste operations and their tendency to cause inconsistencies, a pressing need exists to rethink current editor programs. One solution replaces cut, copy, and paste with operations that correspond directly to the intended semantics behind their use.

Is Production Pulling Knowledge Work to China? A Study of the Notebook PC Industry
pp. 36-42
Jason Dedrick and Kenneth L. Kraemer

During the past 15 years, China has gone from being a minor player in the computer industry to become the world’s largest hardware producer. China began outpacing the US as well as Japan, Singapore, and Taiwan in 1995; since 2000, Chinese hardware production has tripled while output in the other countries has declined.

Yet a 1998 review of globalization in the computer industry indicated that creation of jobs in R&D, product design, software, and IT services more than offset the loss of computer manufacturing jobs in the US. Such activities are also globalizing, however, as companies outsource knowledge work to countries such as Ireland, Israel, India, and now China. To address these issues, the authors studied the globalization of new product development in the notebook PC industry.

Can Indian Software Firms Compete with the Global Giants?
pp. 43-47
Prabhudev Konana

The Indian IT services industry is having a dramatic impact on global consulting services. Giants like IBM, Accenture, EDS, and Deloitte are expanding rapidly in India to benefit from the low-cost, high-quality workforce, while Indian software firms are expanding in the US to create the closer customer relationships required to compete in providing high-end consulting services. This cross-movement of jobs and labor will create an interesting competitive dynamic as the cost structure converges and firms compete to provide knowledge leadership—concept, technology, and process innovation.

To become recognized global brands and move up the value chain, Indian software firms must promote knowledge leadership, create incentives to innovate, and foster efforts to enhance cultural alignment.

On the Credibility of Manet Simulations
pp. 48-54
Todd R. Andel and Alec Yasinsac

Mobile ad hoc networks allow rapid deployment because they don’t depend on a fixed infrastructure. These manet nodes can participate as source, destination, or an intermediate router. This flexibility is attractive for military applications, disaster-response situations, and academic environments where fixed networking infrastructures might not be available.

Manet publications typically include performance simulations and commonly compare routing protocols. The authors show how simulation can systematically produce misleading results and offer a cautionary tale for all who rely exclusively on simulation for validating technical concepts.

pp. 55-62
Yu-Chee Tseng, Meng-Shiuan Pan, and Yuen-Yung Tsai

Wireless sensor network design is generally application driven—that is, a particular application’s requirements will determine how the network behaves. However, previous navigation applications weren’t designed with emergency applications in mind.

The authors propose a distributed navigation algorithm for emergency situations. During normal times, sensors monitor the environment. The authors have developed a protocol that quickly separates hazardous areas from safe areas when an emergency event occurs. Simulation and implementation results show that their scheme achieves navigation safety and quick convergence of the navigation directions.