Evolving significantly during the past two decades, software system creation and maintenance has displayed a synergy of techniques and tools. In fact, recent researchers view the synergy between software engineering and AI (especially expert systems) as the rule rather than the exception.

Two seminal papers—one on software development support for AI programs6 and the other on AI support for software engineering7—led to an FJCC 87 panel focusing on issues concerning "practical and productive interactions between AI and SE." Participating panelists, in addition to authors Tanik and Yun, were Robert Balzer (the Information Sciences Institute), Roger Bate (Texas Instruments), Woodrow W. Bledsoe (the University of Texas at Austin), C.V. Ramamoorthy (UC-Berkeley), and Winston Royce (Lockheed).

In due course, panel members sought to broadcast the emerging synergy between AI and software engineering. The idea received support from Editors-in-Chief David Pessel, Bruce Shriver, and Ted Lewis. Prospective authors responded enthusiastically with numerous high-quality submissions. We have selected the best. The Computer Society is publishing seven of these submissions in companion November issues of Computer, IEEE Expert, and IEEE Software.

IEEE Expert will continue the AI/Software Engineering track with three additional articles in succeeding issues.

In this issue

"Knowledge-Based Support for Rapid Software Prototyping" describes the structure of an expert system for retrieving reusable software components. Luqi focuses on interactive systems for computer-aided prototyping that rely on reusable software components drawn from a software base for constructing prototypes rapidly.

In "KBRA: A New Paradigm for Requirements Engineering," Czuchry and Harris report on KBRA—their knowledge-based requirements assistant—part of a larger Rome Air Development Center effort called KBSA (knowledge-based software assistant).

KBRA provides computer support for formal and informal aspects of requirements acquisition and analysis.

In "Interactive Problem Solving Using Task Configuration and Control," Kant examines a method that supports interactive problem solving with a task-representation-and-control system. She describes a representation for constructing programs incrementally, noting that her system currently supports research projects in program synthesis and mathematical-model design and understanding.

Challenges still exist—for example, identifying the synergistic integration of technological knowledge accumulated in AI, expert systems, software engineering, database management, and graphics. When addressing such challenges, we must maintain a realistic perspective and avoid unrealistic expectations.

On the other hand, we should remember that software engineering is a creative process for computer scientists—much like VLSI-chip design for electrical engineers, or structure engineering for architects. We can enhance the productivity of this process with intelligent programming techniques and system development expertise based on knowledge and experience accumulated within computer science. What could be more appropriate than delivering system development and programming expertise from computer science through AI—a technology developed within the same field? Expert system technology provides an excellent vehicle for delivering domain-specific expertise to users.

References


Murat M. Tanik directs the Software Engineering and Knowledge-Based Systems Research Group and is an associate professor at Southern Methodist University. He is conducting research in knowledge systems and development tools, symbolic and algebraic algorithms and systems, and computational mathematics and software. He received his BS in mathematics from Ohio State University in 1970 and his PhD in applied mathematics and computer science from MIT in 1973. Before joining SMU, he managed IBM T.J. Watson Research Center’s Symbolic and Algebraic Computations Group; was a visiting professor at Stanford, Yale, the Universities of Chicago, Hawaii, and Rome; and was an adjunct professor at NYU’s Courant Institute of Mathematical Sciences. A senior member of the IEEE and a member of the ACM, AAAI, and SIAM, he has published numerous articles in ACM, IEEE, and SIAM journals.

David Y.Y. Yun is a professor and Computer Science and Engineering Department chair at Southern Methodist University. His current interests include knowledge systems and development tools, symbolic and algebraic algorithms and systems, and computational mathematics and software. He received his BS in mathematics from Ohio State University in 1970 and his PhD in applied mathematics and computer science from MIT in 1973. Before joining SMU, he managed IBM T.J. Watson Research Center’s Symbolic and Algebraic Computations Group; was a visiting professor at Stanford, Yale, the Universities of Chicago, Hawaii, and Rome; and was an adjunct professor at NYU’s Courant Institute of Mathematical Sciences. A senior member of the IEEE and a member of the ACM, AAAI, and SIAM, he has published numerous articles in ACM, IEEE, and SIAM journals.

The authors can be reached at the Computer Science and Engineering Dept., School of Engineering and Applied Science, SMU, Dallas, TX 75275-0122.