More than 20 years ago, requirements analysis was identified as a key issue in the success of software development and maintenance. A recent update of the famous Standish Group Chaos study confirms the criticality of requirements engineering processes. Today, many organizations and companies have established explicit roles for requirements engineers. Adequate techniques and tools for RE tasks (such as elicitation, validation, negotiation, specification, and documentation) have emerged and continuously been improved based on industrial feedback.

Since the early 1990s, RE has increasingly been recognized as a discipline of its own. In 1993, the first IEEE International Symposium on Requirements Engineering (RE) took place. This biannual meeting mainly provided a forum for exchanging research results and ideas. In 1994, the first biannual International Conference on Requirements Engineering (ICRE), designated as the field’s technology transfer conference, occurred. Several workshop series, working groups, research networks, journals, and other national and international conferences emerged around the same time.

In 2002, the practice-oriented conference series and the research-oriented symposium series merged—a major milestone toward a mature RE community. The resulting annual IEEE International Requirements Engineering Conference (RE) is now the field’s flagship conference. It provides an excellent forum for establishing relationships and collaborations between industry and research and for discussing and exchanging industrial experiences, problem statements, research results, visions, and, of course, innovations. The first merged conference, RE 02, took place in September 2002 in Essen, Germany.

RE 02 demonstrated the field’s vitality. From over 250 submissions, the program committee selected 27 research papers, 22 experience papers, and 34 industrial talks. Over 300 participants from public and private organizations, labs, and universities (half from industry, half from the research community) experienced strong convergence toward an integrated RE community. The challenges raised and the experiences reported on resulted in numerous fruitful and stimulating discussions within and outside the sessions.

Conference participants identified several emerging themes as well as significant improvements of established techniques and tools, including the following:

Continuous requirements management. Traditionally, RE was seen as an early phase in the system development process. As proposed in the mid 1990s, shorter time to market, technology changes, and frequently changing environments force a shift in this traditional view. So, RE should be understood as a continuous activity that manages requirements evolution throughout the system life cycle and between system boundaries. Several presentations tackled issues...
regarding the deployment of continuous RE processes. In particular, the article by Matthias Weber and Joachim Weisbrod in this issue nicely summarizes their experiences and challenges (both technical and organizational) adopting intensive requirements management processes in several automotive development business units.

Gap between requirements and software. A continuum should exist from the problem domain to the solution domain, as a problem's structure usually differs from that of the intended solution. Some presenters discussed solutions for guaranteeing this continuum based on systematic consistency checks between the two levels. Others emphasized the definition and use of well-understood reusable problem structures (problem frames, requirements patterns) that can then be systematically associated with well-defined architectures.

Managed RE processes. There was clear evidence that RE processes are defined, deployed, and enacted within large and (sometimes geographically) distributed multisite organizations. The article by Stewart A. Higgins, Maurice de Laat, Paul M.C. Gieles, and Emilienne Geurts illustrates the importance of requirements process management at Philips Medical Systems. It reports on the introduction of a centralized, coordinated requirements documentation management system and the supporting processes that it requires.

RE for software product families. This differs significantly from RE for single software products. For example, during the design of a product family's common core assets (domain engineering), we must define variability aspects at the requirements level. This task bears several new challenges. Some presenters emphasized the importance of coordinating the product family domain model with the evolution as well as with product derivation activities.

Return on investment. Several presenters tackled the need to consider ROI at the RE level through requirements management, product families, and other approaches. In particular, they discussed the proactive reuse (recycling) of requirements in products derived from a product family and the directly related increase of ROI. ROI is equally important when deciding about “make or buy”—that is, when to choose COTS products. In this issue, the article by Xavier Franch and Juan P. Carvallo presents an innovative approach for selecting and evaluating software packages based on an ISO-based quality model.

For more details about these and other topics discussed at RE 02, please refer to the conference proceedings.3,4

We are glad to present in this special issue three articles that were selected by the RE 02 program committee and the IEEE Software Editorial Board. All three articles report on the experiences and challenges that industry faces when applying emerging RE approaches.

See you at RE 03 in Monterey, California, 8–12 September 2003 (www.re03.org).

Acknowledgments
Publication of these papers would not have been possible without the reviewers’ valuable and constructive comments. Special thanks to Robert Cochran, Christof Ebert, Suzanne Robertson, and Dale Strok for the smooth editorial process. We also thank the authors, who ably considered the reviewers’ comments when preparing their improved and extended papers.

References

About the Authors
Eric Dubois, program cochair of RE 02, is director of the Center for IT Innovation at the Public Research Center Henri Tudor, Luxembourg, and visiting professor at the University of Namur, Belgium. Contact him at Centre de Recherche Public Henri Tudor, 6 rue Coudenhove Kalergi, L-1359 Luxembourg-Kirchberg; eric.dubois@tudor.lu; www.cit.tudor.lu.

Klaus Pohl, program cochair of RE 02, is director of the Institute of Computer Science and full professor for software systems engineering at the University of Essen. Contact him at the Inst. of Computer Science, Univ. of Essen, Software Systems Eng., Altendorferstr. 97-101, 45117 Essen, Germany; pohl@cs.uni-essen.de; www.sse.uni-essen.de.