RE'03:
Practical Requirements Engineering Solutions

Roel Wieringa, *University of Twente*

Christof Ebert, *Alcatel*

Over the course of the last decade, requirements engineering has evolved into a multidisciplinary field that blends software engineering, systems engineering, product management, and psychology. To put it in precise words, RE is the branch of systems engineering concerned with the desired properties and constraints of software-intensive systems, the goals to be achieved in the software's environment, and assumptions about the environment. It deals with these aspects

of systems engineering from the problem analysis stage to the system implementation and maintenance stages. RE varies greatly depending on the domain involved, ranging from public-administration software to workflow systems, groupware, embedded systems, and control software.

RE interfaces with software engineering in that it specifies the desired functions, quality attributes, and other properties of the software that is to be built or assembled. It interfaces with systems engineering in that it analyzes the software problems that exist in the sociotechnical system in which the software is to play a role. As a problem analysis discipline, RE borrows from product management and psychology; it deals with goals to be achieved, the stakeholders who have these goals, and the problems to be solved within given business constraints. In other words, RE maps needs to solutions. The sidebar describes

Requirements Engineering Tools

Numerous tools are available to support requirements engineering. Here's a small group of well-known ones. Table A lists each tool together with its current supplier, a URL, some key distinguishing features, and an entry-level cost. The difference between low and high cost is at least one order of magnitude. Some low-cost tools offer free licenses. A more complete reference to such tools, with a self ranking of the respective suppliers versus a rich set of requirements, is available at www.incose.org/tools/tooltax.html. A rather subjective but useful summary on tools is available at www.volere.co.uk/tools.htm.

Table A

Well-known RE tools, their suppliers, key features, and entry-level costs

Tool	Supplier	Key features and the tools it integrates with, ranked from the most relevant for the respective purpose	Entry-level cost
Caliber-RM (www.borland.com)	Borland	Lifecycle-oriented, large systems, traceability; TestDirector, Borland Star Team, MS Project	High
CARE (Computer-Aided RE) (www.sophist.de)	Sophist Technologies	Database-like view, requirements-centric; UML tools, MS Office, Lotus Notes	Low
Doors (www.telelogic.com)	Telelogic	Integrated management, large projects, API available, XML support; PLM, UML tools, MS Project	High
IRqA (Integral Requisite Analyzer) (www.tcpsi.es/irqa/ingles/irqa.htm)	TCP Sistemas & Ingeniería	Requirements classification, OO analysis and entity relationship method for database design, traceability, test support, XML support; MS Office	Medium
Reqtify (www.tni-valiosys.com)	TNI-Valiosys	Traceability and impact analysis; text processing, office tools, Simulink	Low
Requisite Pro (www.rational.com)	IBM Rational	Change management, traceability, XML support; MS Word, Rational Rose, TeamTest, MS Project	High
RM Trak (www.rmtrak.com)	RM Trak	Entry-level requirements management; MS Office, SQL interface	Low
RTM Workshop (www.chipware.com)	Integrated Chipware	Multiuser distributed projects, modeling, API available, information modeling; UML tools, test tools, SQL	High
Truereq (www.truereq.com)	Truereq	Lifecycle-oriented, team-centric, entry level, XML support; Web browser forms	Low
Vital Link (www.complianceautomation.com)	Compliance Automation	Database-centric, large projects, SQL, API available; Adobe Framemaker	Medium

several popular RE tools, without claiming to be exhaustive.

The RE conference series (www.requirements-engineering.org) is a platform for the industrial community to present relevant experiences and for the research community to present novel results. The series started in the early 1990s as two separate IEEE-sponsored conferences but merged in 2002. The 2003 conference took place in beautiful Monterey, California.

Today's trends

The meeting brought forward several important new trends. One trend is the rising interest in RE for commercial-off-the-shelf systems. In both administrative and technical automation, the days are long past when programmers would write their own solutions to customers' problems from scratch. Today, solutions are composed from various components provided by third parties or reused from other systems. Often different suppliers cooperate in integrating a solution for a given contract. Consequently, solution specifications must deal with the fact that the available solution components might not readily have the properties needed to address the customer's needs in the best possible way. RE is then a process of matching known problems and known solution elements, rather than of reasoning from problem to solution. To address such issues, a new RE'03 workshop

About the Authors
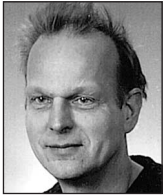
Roel Wieringa, program chair of RE'03, is full professor of information systems at the University of Twente, The Netherlands. He also chairs the Steering Committee of the IEEE International Requirements Engineering Conference. Contact him at the Dept. of Computer Science, Faculty of Electrical Eng., Mathematics and Computer Science, Univ. of Twente, PO Box 217, 7500 AE Enschede, Netherlands; roelw@cs.utwente.nl; www.cs.utwente.nl/~roelw.

Christof Ebert is director of software coordination and process improvement at Alcatel, where he drives R&D innovation and improvement programs. His responsibilities include leading Alcatel's R&D to CMM Level 4. He also lectures at the University of Stuttgart, Germany, on real-time systems. He received his PhD in electrical engineering from the University of Stuttgart. He is a senior member of the IEEE and *IEEE Software*'s associate editor in chief for requirements. Contact him at Alcatel, 54 Rue La Boetie, 75008 Paris, France; christof.ebert@alcatel.com.

series began called COTS and Product Software: Why Requirements Are So Important—RECOTS for short.

A second trend is the growing interest in requirements engineers' skill sets. In the context of large systems, a requirements engineer must be able to identify and understand problems related to policy planning and business strategy, marketing and finance, systems integration, and product development. Academic programs alone can't create and shape these skills. They must be acquired through years of practice and reflection on effective practices in various contexts. Many conference goers attended Ralph Young's tutorial on RE skills, and keynote speaker Heinz Stoewer, then-president-elect of the International Council on Systems Engineering (INCOSE), presented a systems engineering view of evolving RE skills.

A third trend we observed is the wish to learn and to share experiences on effective ways to implement RE in an organization. During a lively panel on this topic, researchers expressed concern about the lack of RE in some industrial projects, and managers expressed their concerns about the need for more practical RE research results. The ensuing discussion proved illuminating for both sides and continued informally during the entire conference.

Along those lines, the RE community is increasingly interested in sound RE research methodology. The standard criticism that a research result is "all right in theory, but it won't work in practice" might be justified in some cases, but it doesn't lead to improvements. Often in such cases, someone is applying theory to a situation where it's inapplicable—it's probably a good solution, but to another problem. According to Abraham Kaplan (*The Conduct of Inquiry*, Transaction Publishers, 1998), nothing is as practical as a good theory. But this means that the conditions under which a theory applies must be clearly expressed. This in turn requires attention to sound research methodology. The RE conference and the articles in this special issue serve as a way to translate research results and industry experiences into advances in RE methodology.

The articles

We chose three papers from the RE'03 conference for this special issue of *IEEE Software* because they address practical needs effectively while building on sound research results.

"Ongoing Requirements Discovery in High-Integrity Systems" by Robyn R. Lutz and Inés Carmen Mikulski presents the results of empirical research on the processes by which new requirements for high-integrity systems software are discovered. The article also does away with the myth that RE takes place before a system is launched—in fact, it occurs over the system's entire life.

"ERP Requirements Engineering Practice: Lessons Learned" by Maya Daneva illustrates RE's importance for a particular class of COTS software, namely enterprise resource planning systems. The author builds on a rich base of experience implementing these systems, in which she matches the requirements, which are based on business process and data needs, with the functionality actually available in existing ERP systems.

"Executable Use Cases: Requirements for a Pervasive Health Care System" by Jens Bæk Jørgensen and Claus Bossen introduces a new tool: executable use cases. Requirements for pervasive software systems are notoriously hard to elicit because stakeholders must rely on their imaginations to envision the opportunities a novel technology offers. The authors combine specifications of use cases in natural language, formal models, and animation to explore these systems' requirements.

These articles from RE'03 represent just a bit of the valuable knowledge the RE community can offer. We hope you enjoy them. RE'04 (www.re04.org) will take place in Kyoto, Japan, on 6–10 September.