*Journal of Universal Computer Science, vol. 10, no. 4 (2004), 303-305*
*submitted: 1/4/04, accepted: 10/4/04, appeared: 28/4/04* © *J.UCS*

# Breakthroughs and Challenges in Software Engineering J.UCS Special Issue

**Rafael Corchuelo**
(Universidad de Sevilla, Spain
corchu@us.es)
**Antonio Ruiz-Cortés**
(Universidad de Sevilla, Spain
aruiz@us.es)

In the seminal 1968 NATO Conference on Software Engineering, F.L. Bauer defined this term as "the establishment and use of sound engineering principles in order to obtain software that is reliable and works efficiently on real machines". In the beginning, most proposals focused on programming languages and techniques, which are currently in a rather elaborate state. Later, the discipline experienced a shift from programming to project management, requirements, analysis, design, or maintenance issues. Furthermore, the Internet seems to be a new cornerstone that is gaining currency and paving the way for a new generation of software applications that are becoming more complex at an ever increasing pace. One of the reasons lies in the heterogeneous nature of the run-time and development-time aspects to be taken into account: attractiveness, quality of service, security, robustness, distribution, standards, and so on. Therefore, Software Engineering needs to keep producing methods and tools to tackle the increasing complexity of next-generation applications, and there is still place for research on methods and tools for project management, requirements elicitation, software analysis and design, testing and verification, or new supporting technologies.

The goal of this special issue was to collect high-quality articles on Software Engineering breakthroughs and challenges. Our first attempt focused on the Spanish Conference on Software Engineering and Databases, which is a mature local conference in which researchers from Spain and surrounding countries meet yearly to share experiences and report on their latest findings. However, we thought that the issue would benefit from contributions around the world, and we decided to invite a number of renowned authors. All in all, we selected eight papers out of sixty five excellent submissions from Australia (5%), Canada (3%), Spain (60%), France (3%), Germany (3%), India (1%), Italy (4%), Japan (1%), Portugal (7%) Korea (4%), Singapore (2%), the United Kingdom (3%), and The United States of America (4%). The selection process was difficult since most articles seemed to be quite good. Only thirty articles passed the first reviewing round, of which we could select for publication only eight, namely:

- The first article reports on requirements negotiation, and it was written by Hoh Peter In, from the Korea University, and David Olson, from the University of Nebraska. Despite the wealth of state-of-the-art development tools available today, there are still many software projects that end up in disaster. It is widely accepted that many such projects begin before a precise requirements specification is negotiated. In this article, the authors propose a new requirements negotiation framework that builds on multi-criteria preference analysis and has proved to be successful in an industrial/academic experience in which complex requirements to build a university library repository system were negotiated.
- The second article deals with method engineering, and it was written by Brian Henderson-Sellers, Magdy Serour, Tom McBride, Cesar Gonzalez-Perez, and Lorraine Dagher, from the University of Technology, Sydney, Australia. The authors argue that methodological proposals that rely on the common idea that one size fits all, are prone to fade away quickly and to be untenable. They advocate the use of so-called Situational Method Engineering, which is a proposal in which different methods are stored in a repository and combined as necessary to prepare methodologies that adapt smoothly to different projects at different capability levels. Their results are illustrated by means of a couple of detailed examples.
- The third article was written by Jorge Gómez-Sanz and Juan Pavón, from the Complutense University of Madrid, Spain. Software agents are a promising software engineering approach for building complex distributed systems in which many autonomous artefacts interact, negotiate, compete, and work to perform complex tasks on behalf of human users or other computer-based systems. It is not surprising then that many researchers focus on methodologies to build such systems. In this article, the authors have made a great effort to describe current state-of-the-art methodologies to build multi-agent systems from an eclectic, yet rigorous point of view.
- The fourth article focuses on coordinating web services and it was written by Luís Filipe Andrade, who works for ATX Software S.A., Portugal, and José Luiz Fiadeiro, from the the University of Leicester, United Kingdom. They report on recent coordination challenges raised by the emergence of web services. They argue on that web services cannot be coordinated using common object-oriented coordination tools since they are far more dynamic and prone to run-time changes than objects. They proof that composition laws and interfaces can be used so that the behavior of a business process can be described in terms of interactions with given partners.
- The fifth article, which was written by Roger T. Alexander, from Colorado State University, USA, and Jeff Offutt, from George Mason University, USA,

reports on testing. Researchers have put a great deal of effort into implementing methods to test software applications in an attempt to find bugs before they are released and deployed. Most efforts have focused on procedural programs, which are radically different from object-oriented programs. Most difficulties arise due to polymorphic relationships that make it difficult to know which the actual type of a reference is. In this article, the authors describe a few techniques to analyse and test such relationships that help increase the ability to find bugs.

- The sixth article reports on migrating legacy software systems, and it was written by Artur Boronat, Jennifer Pérez, José Á. Carsí, and Isidro Ramos, from the Politechnic University of Valencia, Spain. They have been working on legacy systems for some time, and they have developed a couple of tools that help software engineers migrate their databases easily. Their proposal builds on an algebraic language that allows to represent conceptual schemes and produce migration plans automatically. The tools were tested on industrial projects and proved to reduce migration costs by orders of magnitude.

- The seventh article goes deeper into web challenges. It was written by Óscar Díaz and Juan J. Rodríguez from the University of the Basque Country, Spain. They report on the so-called portlet technology. Portlets constitute a common technique to reuse pieces of functionality that is offered though a web portal and was not designed to be reused in other applications. Unfortunately, there is not an agreed portlet model, which makes interoperability a challenge. The authors report on these problems and present several implementation patterns.

- The eighth article was written by Mercedes Amor, Lidia Fuentes, and Mónica Pinto, from the University of Málaga, Spain. The authors have been working on multimedia systems for years, so they first present a detailed report on what challenges a multimedia programmer has to face. Building on their experience, they have developed a couple of frameworks to help programmers incorporate multimedia information into their applications. They have made an effort to present the benefits of applying component-based software development and application framework technologies.

Last, but not least, we would like to thank our reviewers for their hard work and enthusiasm to produce this special issue, and the J.UCS staff for their help and understanding. Without a shadow of doubt, they were fundamental to producing this issue, and we would not like to miss this opportunity to express our gratitude.

Rafael Corchuelo, Antonio Ruiz-Cortés
(Sevilla, April, 2004)