Today, we live in a world in which our safety is more and more dependent on software-intensive systems. This is the case for the aeronautic, automotive, medical, nuclear, and railway sectors, as well as many more. Organizations everywhere are struggling to find cost-effective methods to deal with the enormous increase in size and complexity of these systems, while simultaneously respecting the need to ensure their safety. Consequently, we’re witnessing the ad hoc emergence of a renewed
OPEN PLATFORM FOR EVOLUTIONARY CERTIFICATION OF SAFETY-CRITICAL SYSTEMS

Safety-critical software faces a costly aspect: the certification process. OPENCOSS, a large-scale collaborative project of the EU’s Seventh Framework Program, focuses on the harmonization of safety assurance and certification management activities for the development of embedded systems in automotive, railway, and aerospace industries. The main goal is to reduce both the time and cost overheads inherent to the safety (re) certification of safety-critical systems, via facilitating the reuse of certification assets. The strategy is to focus on a compositional and evolutionary certification approach with the capability to reuse safety arguments, safety evidence, and contextual information about system components in a way that makes approvals for operation and certification more cost-effective, precise, and scalable.

OPENCOSS is defining a common certification language (CCL) by unifying the requirements and concepts of different industries and building a common approach to certification activities. Much of what is being done will have a transformative effect on the safety-critical software community if the take up really occurs. An industrial adoption program is being overseen by an advisory board with members from key organizations such as the European Railway Agency, Airbus, Eurocopter, NASA, and Renault. For more information, see the website: www.opencoss-project.eu.

disciplinary of safety-critical software systems development as a broad range of software engineering methods, tools, and frameworks are revisited from a safety-related perspective. A major goal of this special issue of IEEE Software is to take stock of these individual initiatives and try to see the bigger picture.

Complexity Scales

As an example of important paradigms currently being revisited in a safety-related context, Thales recently announced the use of object-oriented technologies and agile software development methodologies to optimize its safety-critical systems development (www.ertss2012.org/Site/0P2RUC897A/-4. pdf). Likewise, NASA is exploring the study and application of agile development in its safety-critical systems (http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120013429_2012013093.pdf).

But it isn’t just the popular, headline-grabbing software engineering techniques such as agile development that are being revisited in the safety-critical systems community. Understanding the effects of fundamental software engineering activities, including verification, validation, and certification, and choosing the right combination of them to yield systems that meet today’s ambitious requirements in a cost-effective manner has become even more important. Consider the requirements engineering activity: How is it possible that, given the crucial importance of clear, concise, unambiguous requirements in critical software systems engineering, most tools in common use today still represent a requirement as a simple, unadorned string? The European Space Agency’s recent study on next-generation requirements engineering, in which it used semantic wiki technology to nudge forward the state of the art, is just one example of the critical software community’s growing impatience with traditional methods.

Several new and unprecedented factors are converging to change the nature of the challenges facing safety-critical systems development. One such factor is the unrelenting trend toward open, interconnected, networked systems (such as “the connected car” and the cloud), which has brought a security dimension with it, exacerbating the problem of ensuring safety in the presence of security requirements. Similarly, the model-driven architectures (such as AUTOSAR in the automotive industry) needed to handle these new large, networked systems are only now being equipped with mechanisms to handle safety-related aspects. The rise of these complex, critical systems has spawned several recent initiatives to promote reuse, both of the technical artifacts and the artifacts and procedures that certify their suitability for use in safety-related contexts. An example of such an initiative is OPENCOSS, an all-out, full-frontal assault on managing the problem of certification of software-intensive critical systems in multiple domains using model-based approaches and incremental techniques (see the sidebar).

In This Issue

This special issue collects three papers from academia, two from industries, and two from academia with an industrial perspective. This balance provides a rather complete view of the current challenges faced in safety-critical industries despite the specific transportation industries represented. Model-based development and engineering is discussed in "Model-Based Development and Format Methods in the Railway..."
Industry,” “Validating Software Reliability through Statistical Model Checking,” and “Engineering Air Traffic Control Systems with MDE.”

These articles address the challenges and failed expectations in applying these techniques, and highlight the missing link between academia and industry regarding this topic and the importance of tools to support implementation. We thank the authors of these three articles for providing real examples on how to deploy these techniques and believe that their expertise can be reused. “Testing of Formal Verification,” based on DO-178C, is another easy-to-read article that digs into the attractiveness of formal methods technology for high-integrity systems. It’s important to look at the trends in that domain, especially when two major aircraft manufacturers—Airbus and Dassault-Aviation—report the benefits realized.

This issue includes two other articles describing real cases as well. The article from Moog India Technology Center—another aircraft player—provides a collection of mistakes made and their root causes; the focus is on the numerous interactions the aircraft or flight system has with embedded systems that make certification of these systems so complex. “Strategic Traceability for Safety-Critical Projects” likewise targets the traceability issue, which is one of the key facets of certification; the authors provide a fairly detailed analysis of a few traceability issues and the way they were corrected.

Although embedded systems generally come to mind first when thinking of safety-critical software, another class of applications is equally important: the protection of the infrastructures that are critical to our everyday lives, such as transport systems. Although threats usually come from nature, such as hurricanes, earthquakes, and rainstorms, some threats are man-made, such as terrorism and sabotage.

The software systems that protect these infrastructures must span international borders and bring a host of technical, legal, and cultural compatibility challenges with them that in many respects equal or surpass those faced in critical embedded systems. The last article of this issue, “SCEPYLT: An Information System on Explosive Control” provides insight into the issues faced by this type of critical system.

One unmistakable trend that emerges out of the articles in this special issue is a strong interest in applying model-driven engineering techniques to safety-critical systems development over the entire life cycle. The implementation community has been interested in model-based techniques for years, but the validation and certification community is slowly coming around to a perception that such approaches could provide the key to more efficient and effective management of their own tasks. We believe that this observable transition of a research technique into an industrial environment in which certification bodies are neither system nor software technology specialists is a significant step forward in safety-critical systems engineering and an interesting achievement to be reported in this magazine.

XABIER LARRUCEA is a senior project leader at Tecnalia, Zamudio, Spain. He’s also a part-time lecturer at the University of the Basque Country. His research interests are focused on safety-critical software systems, software quality assurance in multimodel environments, empirical software engineering, and technology road mapping. Larrucea has a PhD in software engineering from the University of the Basque Country. Contact him at xabier.larrucea@tecnalia.com.

ANNIE COMBELLES is the founder and CEO of inspearit, an advisory company in software and systems operating in France, Holland, Italy, and Asia. She’s an associate editor of this magazine, a member of the Scientific Committee for Quality Engineering Laura Basai Lab (QE LaB) in Austria, and a member of the executive committee of Les Journées de l’Entrepreneur. Contact her at annie.combelles@inspearit.com.

JOHN FAVARO is a senior consultant at Intecs SpA in Pisa, Italy, where he’s also deputy director of research. His technical interests include efficient safety analysis of critical systems, safe and secure software reuse, and requirements engineering. Favaro has an MS in electrical engineering and computer science from the University of California, Berkeley. Contact him at john@favaro.net.