Innovations Syst Softw Eng (2009) 5:95–96
DOI 10.1007/s11334-009-0083-6

EDITORIAL

# Special section on real-time safety-critical systems

**Janusz Zalewski**

Received: 15 March 2009 / Accepted: 20 April 2009 / Published online: 5 May 2009
© Springer-Verlag London Limited 2009

Real-Time Systems is the domain of computing, which deals essentially with computer systems whose operation is subjected to bounded response time. Related timing restrictions may be soft, such as in airline reservation systems, or hard, as in flight control systems or anti-lock brakes. Hard real-time systems, such as those mentioned, are additionally safety-critical, which means that their failure may cause loss of life, injuries, or large financial losses. Thus, in the design and development of such systems special techniques should be used that ensure not only responsiveness and timeliness of the system in operation, but also guarantee their safety, to the highest extent possible.

Designing real-time safety-critical systems has always been a challenge and a number of special issues of various journals and conference proceedings addressed related problems over the recent years. In this Special Section we are focusing on the design level of system and software development, as opposed to the implementation level. At this level, we are concerned about providing assurance in the design that the system will meet its timing and safety requirements. There are two traditional approaches to address this issue: an engineering approach, which relies on using templates and proven patterns, considering their pros and cons under various operating conditions, and a formal approach, which relies on using techniques of discrete mathematics with its theorem proving apparatus to assist in addressing system safety assurance in the development process.

Papers in this section are extended versions of those presented at the RTS'08 International Workshop on Real-Time Software, held in Wisła, Poland, on 20 October 2008, as a part of the IMCSIT'09 International Multiconference on Computer Science and Information Systems.

J. Zalewski (✉)
Florida Gulf Coast University, Fort Myers, USA
e-mail: zalewski@fgcu.edu

The engineering approaches have been much more successful in real-life applications, thus far, because of their development base and years of documented experience. Nevertheless, there are still new challenges in emerging applications, mostly due to the increasing complexity of systems and miniaturization of devices. Two papers in this issue are addressing this subject. First, Gumzej and Halang advance the concept of a safety shell, co-developed by the author of this Editorial, to use it in a safety pattern of S-PEARL specification and programming language, which can be used in real-time applications developed with UML-RT. The second paper, by Sterritt and Hinchey, describes several developments for self-managing real-time systems, forthcoming in future NASA missions for space exploration. Such missions will certainly require the use of advanced and non-traditional concepts and technologies, and the paper gives us a probable view of what real-time and safety-critical computing may look alike in the future.

Three papers on the use of formal methods nicely enhance the discussion of engineering issues by a more mathematically oriented presentation. The first paper, by Furfaro and Nigro, is a good transition between the two approaches, as it discusses a real-time extension of the well-established engineering approach, DEVS (Discrete Event System Specification), and the process of transformation of its specifications into UPPAAL for exhaustive verification based on model checking. In another paper, Martin Kot is also using UPPAAL for modeling concurrency control in real-time database management systems. On the other hand, a paper by Rysavy and Rab presents an approach to modeling real-time embedded systems based on yet another formal method known as TLA+ (Temporal Logic of Actions).

The final paper in this Special Section, by Kornecki and Zalewski on certification issues in real-time safety-critical software, discusses the state-of-the-art in this important area

of computing. Software, which controls embedded devices and industrial processes that affect our lives, must be proved to be safe, that is, free from faults that may cause its malfunctioning and, as a consequence, negatively impact the entire system and contribute to injuries or deaths of human beings. Developing methods, technologies and standards that help us verify that the software is "fit for purpose" is of significant importance to the society as a whole. Thus, the discussion of engineering and formal approaches to software design ends with this overview.