Computer-based control systems are by no means simple things to be designed and implemented, as they are not only involved with interfacing and real-time computing issues, but also face problems related to fault tolerance, implementation constraints, and, not least, particular designers’ attitudes. As a consequence, increasing attention is being paid to devising methodologies and developing tools to reduce the designers’ efforts and to increase the quality of the implemented systems. The four papers of this session can be considered samples of a specialized discipline emerging as a special thread inside the mainstream of software engineering.

The first paper is *A Macro Expansion Approach to Embedded Processor Code Generation*, by E. Lassila. Assembly language is still the approach of choice for many embedded systems implementations, especially when based on special-purpose processors. In spite of its low level, this programming practice should, and can, be carried out paying attention to structure and modularity, and some tools are worth the efforts to make the programmer’s life as easy as possible. The paper suggests macro expansion as a way to accomplish this purpose, while pursuing code optimization through program flow analysis.

The second paper, *Functional Validation of Fault-Tolerant Asynchronous Algorithms*, by J. Hlavicka, S. Racek, and P. Šmrha, addresses the problem of validating resilient system behavior in distributed environments with asynchronous events. A simulator is suggested as a tool for experimental validation. To keep the tool as easy to understand and as close as possible to the final implementation, an extension of the C language, by means of a function library, has been developed. The use of the method is illustrated with a simple example.

In the third paper, *A Prototyping technique with an Asynchronous Specification Language*, by M. Švéda, rapid and low-cost prototyping is suggested, as an interesting
way of attacking complex information systems development. Things are more complex when dealing with time-sensitive embedded systems. Along the way, the presented technique makes use of attribute grammars to specify source language constructs, macros to implement syntactic and semantic rules, macroprocessor or Prolog-driven expansions into target language code, to build up a simple prototyping tool.

The last paper, *Software Engineering in Control Using Objects and Services*, by O. Hammerschmidt and T. Doersam, discusses from an engineering perspective many issues and problems faced by the designers of distributed systems for real-time process control. An object-oriented / service-oriented approach is presented, with an example, taking into account both hardware and software aspects of the implementation, and aiming at fast prototyping and automated code generation.