Model-based Engineering of Embedded Real-time Systems

Holger Giese, Gabor Karsai, Edward Lee, Bernhard Rumpe, Bernhard Schätz

Today, embedded software plays a central role in most advanced technical systems such as airplanes, cell phones, and cars, and has become the main driver for innovation. Development, evolution, configuration and maintenance of embedded and distributed software nowadays often are serious challenges as a drastic increase of the software complexity can be observed in practice. The application of model-based engineering technologies to embedded real-time systems seems to be a good candidate to tackle some of the resulting problems.

Model-based development strategies and automatic code generation are becoming established technologies on the functional level. However, they are mainly applied in monolithic systems. The use of similar modeling strategies on the system, technical, and configuration levels remains challenging, especially with the increasing shift to networks of systems, deepened interaction between control-engineering and reaction-oriented parts of a system, and the growing number of variants introduced by product lines. Specific domain constraints such as real-time requirements, resource limitations and specific hardware dependencies often impede the acceptance of standard high-level oriented modeling techniques and their model-based application. Much effort in industry and academia therefore goes into the adaptation and improvement of object-oriented and component-based methods and model-based engineering that promise to facilitate the development, deployment, and reuse of software components embedded in real-time environments. The model-based development approach for embedded systems and their software proposes application specific modeling techniques using domain specific concepts (e.g., time-triggered execution or synchronous data flow) to abstract from the details of the implementation such as interrupts or method calls. Furthermore, analytical techniques (like, e.g., verification of the completeness of function deployment and consistency of dynamic interface descriptions) and generative techniques (e.g., automatic schedule generation, default behavior generation) can then be applied to the resulting more abstract models to enable the efficient development of high quality software.

Our Dagstuhl seminar brought together researchers and practitioners from the field of model-based engineering of embedded real-time systems. The topics covered included: frameworks and methods, validation, model-based integration technology, formal modeling of semantics, fault management, concurrency models and models of computation, requirements modeling, formal derivation of designs from requirements, test modeling and model-based test generation, quality assurance, design management, abstractions and extensions, and development techniques and problems of application domains. The broad spectrum of presentations has clearly illustrated the

prevalence of model-based techniques in the embedded systems area, as well as progress in the field.

The seminar included mostly conference-like presentations followed by short discussions, and three group discussion sessions with panels selected from the attendees. In all cases, the emphasis was on fostering interaction among the participants and on gaining new insights and better understanding. Most of the seminar participants provided abstracts and the slides for their presentations that are available on the Dagstuhl website for the seminar. We plan to compile a state-of-the-art survey on model-based development of embedded real-time systems addressing foundational issues, language engineering, domain-specific issues, and life-cycle issues. The survey will be based on contributions of the participants of the seminar and will be published in the Springer LNCS series.