Embedded Systems - Challenges and Work Directions

Joseph Sifakis

Verimag and ARTIST2 European Network of Excellence

Abstract. Embedded Systems are components integrating software and hardware jointly and specifically designed to provide given functionalities. These components may be used in many different types of applications, including transport (avionics, space, automotive, trains), electrical and electronic appliances (cameras, toys, television, washers, dryers, audio systems, cellular phones), power distribution, factory automation systems, etc.

Their extensive use and integration in everyday products marks a significant evolution in information science and technology. A main trend is the proliferation of embedded systems, that should work in seamless interaction while respecting real-world constraints.

Embedded systems have a number of specific characteristics, which play a role in structuring the technical domain including criticality, reactivity and autonomy.

The coming generations of embedded systems - primarily used in mass-market products - need development methods and tools allowing to jointly consider functionality, quality, physical implementation, and market constraints: The need to jointly consider functional and extra-functional constraints leads to a system-centric approach to development. Here, the main focus is the end result: a system as the combination of hardware and software, in interaction with its physical environment.

Current methods and tools do not allow system-centric approaches. These approaches raise difficult, fundamental research problems, which are the basis of an emerging theory that should bring together information and physical sciences. Information sciences consider models of computation based on abstract notions of machines (e.g., automata, complexity and computability theory, algorithms, etc.), that do not take into account physical properties of computation (e.g., execution times, delays, latency, etc.). There is no unified theory allowing to predict the behavior of an application software on a given execution platform which determines execution speed and other dynamic properties of the application.

System-centric approaches raise two grand challenges common to all the activities of system development. The first is theory and tools for rigorous component-based engineering. This determines our ability to build complex systems from simpler ones by mastering their complexity. The second is intelligence, a long term vision for systems that are able to analyze and adapt their behavior according to changes of their environment.

T. Higashino (Ed.): OPODIS 2004, LNCS 3544, pp. 184–185, 2005.

We discuss specific work directions in system development activities to meet these challenges, including modeling, programming and compilation, operating systems design, controller synthesis, testing and verification.

Reference

1. ARTIST: "Selected Topics in Embedded Systems Design: Roadmaps for Research", http://www.artist-embedded.org/Roadmaps/