Demonstration Acronym:
RUMORE

Demonstration Title:
A Framework for RUntime MOnitoring and TRacE Analysis for Component-based Embedded Systems Design Flow

Authors:
Luigi Pomante¹, Vittoriano Muttillo¹, Giacomo Valente¹, Hector Posadas², Javier Merino², Eugenio Villar²
{luigi.pomante, vittoriano.muttillo, giacomo.valente}@univaq.it, {posadash, javierm, villar}@teisa.unican.es

¹ Department of Information Engineering, Computer Science and Mathematics (DISIM), University of L’Aquila, L’Aquila, Italy
² Department of Electronics Technology, Systems and Automation Engineering (TEISA), University of Cantabria, Santander, Spain

Demonstration Abstract (max 1000 characters):
In the last years, the spread and importance of embedded systems are even more increasing, but it is still not yet possible to completely standardize and engineer their system-level design flow. The main design problems are to model functional (F) and non-functional (NF) requirements and to validate the system before implementation. In such a scenario, monitoring systems have an important role in verification and validation steps, while providing information about system state, that can be processed (e.g. filtered, interpolated, etc.) to obtain indications about parameters (e.g., workload characterization, debug action), or characterize specific components (e.g., cache memories, buses, etc.).

In such a context, the purpose of this demonstrator is to introduce runtime monitoring infrastructures and to analyze trace data. The goal is to show the concept among different monitoring requirements by defining a general reference architecture that can be adapted to different scenarios. Starting from design artifacts, generated by a system engineering modeling tool, a custom monitoring system infrastructure will be presented. This sub-system will be able to generate runtime artifacts for runtime verification. We will show how the RUMORE framework provides round-trip support in the development chain, injecting monitoring requirements from design models down to code and its execution on the platform and log data back to the models, where the expected behavior will then compared with the actual behavior. This approach will be used towards optimizing design models for specific properties (e.g. for system performance). A custom HW monitoring infrastructure for reconfigurable logic architectures will be presented, with a focus on Zynq platform and FPGA boards. HW monitoring systems will be introduced into a reconfigurable platform, and a general trace data extraction framework will be considered. Then, bare-metal and Linux APIs will provide run-time information from HW sniffers. A transformation toward Common Trace Format (CTF) will be proposed for runtime validation activities in order to check input constraints. Results targeting real case studies of run-time verification will be presented, together with the projection of CTF on visualizer tools. Finally, trace analyzer tools able to validate high-level component-based timing constraints will be shown, with a focus on data consistency, task dependency flow, and end-to-end component chains timing behavior.