High Level Simulation of Cyber-Physical Systems

Ralph Weissnegger[1], Markus Pistauer[1], and Christian Steger[2]

[1] CISC Semiconductor GmbH, Klagenfurt, Austria r.weissnegger@cisc.at, m.pistauer@cisc.at
[2] Graz University of Technology, Austria, steger@tugraz.at

Summary. This abstract presents the simulation based design and verification of Cyber-Physical Systems (CPS) from a system point of view.

1 Abstract

The rapidly developing application of Cyber-Physical Systems (CPS) in our daily life is driven by a closer integration of computational systems (the cloud) and local devices like sensors and actors networked with each other. This results is a novel class of applications that can execute complex system tasks flexibly autonomously and even allow optional individual user interaction. The design and development of such systems requires a large range of expertise and technological components, spanning from heterogeneous sensor elements and communication systems over scalable data processing systems to physical objects involving also humans as users.

The electrical architecture of a vehicle can be seen as such, although in the past this was not the case. Recent trends in the in-vehicle electrical/electronic architectures brought a rapid shift towards multicore, heterogeneous, networked and reconfigurable systems. This makes the design of such systems extremely complex and imposes a large effort for designers (hard- and software) to design their applications. The next generation of such systems should be able to allow applications to run in parallel on different parts (electronic control units (ECU) and/or processors, DSPs within a multicore architecture of a single ECU) of the system such as Multimedia, Human Machine Interface (HMI) like car infotainment or nomadic device (Tablet, Smart Phone) ad-hoc interconnect, Navigation, Advanced Driver Assistance Systems (ADAS) and many more. This has been acknowledged also on a European Industry Level [1] [2].

Beside performance, power consumption and of course safety relevant aspects have to be considered as well once a certain Automotive Safety Integrity Level (ASIL) needs to be applied.

To guarantee the safety of these systems, they have to follow standards like the ISO 26262 [3]. The ISO 26262 standard is an adaption of the functional safety standard IEC 61508 for automotive electric/electronic systems. Today, the compliance of the standard is optional but is more often required from the automotive-companies and their partners.To be ISO 26262 conform it's important to proof the whole system towards its safety goals, not a set of specialized parts. As a result of the foregoing, such a CPS realized in cars demands very fast executable specifications to validate the system concepts and proof the system with particular emphasis on safety relevant criteria.

System level modeling is nowadays a commonly used methodology in design and verification of complex embedded real-time systems to reduce engineering risks through an early, fast and flexible method to detect design flaws. Designs can be verified and simulated long before committing to its implementation. But most of the engineers involved in these tasks apply EDA tools like e.g. Matlab/Simulink [4]. This approach is direct and effective. On the other hand most details of the realized system, code generation for the target hardware platform, parameter descriptions etc. are linked with a dedicated hardware implementation and unlinked with the global system requirements.

As a consequence a paradigm shift will be necessary to further abstract the system modeling languages applied to simulation and verification of CPS. A Unified Modeling Language (UML [5]) is a language that might be considered for modeling of application models.

Further the language SYSML [6] can be considered as extension/subset of UML to better address the consideration of requirements and the description of system parameters.

Another approach is MARTE [7] as domain-specific modeling language intended for model-based design and analysis of realtime and embedded software of cyber-physical systems. MARTE is an extension to UML, providing additional parameters for real-time systems, which are missing in UML. Non-functional properties are very important to evaluate the functional safety of the system. With MARTE it is possible to map functional (structural and behavioral) and non-functional (power, thermal, time) properties to models. With MARTE Allocation it is possible to associate the functional application elements with the available resources (execution platform). For interoperability with other tools and the use of standardized

IP-models, the open standard IP-XACT [8] can be used.

1.1 Model-to-text transformation

For the simulation of system-models based on MARTE, the high level language code is generated through a model-to-text transformation process. One of the languages used for this approach is SystemC [9]. SystemC is gaining increasing attention because of its great flexibility in describing the components at different levels of abstraction, from system and transaction level down to RTL level. Even extensions to describe analog and mixed signal systems are available. A second important SystemC extension is SystemC-TLM to describe transaction driven sytems. This allows a faster modeling and simulation time while developing phase, to detect HW/SW integration issues. Another language used in this approach is Matlab. Matlab/Simulink is well established in simulation of systems in different domains and used by many other tool vendors but shall not be used in the future for system level descriptions.

1.2 Simulation based verification

To test the system towards their safety goals, testbenches are automatically generated from the safety functions defined in the ISO 26262 process. An automatic stimulus generator and a protocol checker are used to verify the system. The testbenches support transaction level modeling and are written in UVM [10] (Universal Verification Methodology).

Acknowledgement. The approach presented above is experimented in the Catrene (OpenES) research program supported by the FFG (Austrian Research Promotion Agency) in cooperation with the technical University of Graz.

References

1. ECSEL. *2014 ECSEL Multi Annual Strategic Research and Innovation Agenda (MASRIA).* http://www.smart-systems-integration.org/public/documents/publications/2014_ecsel_masria_partc.pdf.
2. Gartner. *Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020.* http://www.gartner.com/newsroom/id/2636073.
3. ISO 26262. *ISO 26262-1:2011(en), Road vehicles Functional safety Part 1: Vocabulary.* https://www.iso.org/obp/ui/, Publication date 2011-11-14.
4. MathWorks. *Matlab R2014a Documentation.* http://www.mathworks.de/de/help/simulink/.
5. OMG. *Unified Modeling Language, Version 2.5.* September 2013, OMG Document Number: ptc/2013-09-05.
6. OMG. *System Modeling Language, Version 1.3.* June 2012, OMG Document Number: formal/2012-06-01.
7. OMG. *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems, Version 1.1.* June 2011, OMG Document Number: formal/2011-06-02.
8. IEEE. *IEEE Standard for IP-Xact, Standard Structure for Packaging, Integrating, and Reusing IP within Tool Flows, Version 1.1.* February 2010, IEEE, 3 Park Avenue New York, NY 10016-5997, USA.
9. Accellera. *SystemC 2.3.1 (Includes TLM).* http://www.accellera.org/downloads/standards/systemc/.
10. Accellera. *Universal Verification Methodology (UVM) 1.1 Class Reference.* June 2011, Accellera Organization 1370 Trancas Street 163 Napa, CA 94558 USA.