

## C4 Model-Based Design of Embedded Systems

Time: Tuesday, 14.09.2010

Location: Humboldt-Building, Lecture Room 129

Chairman: A. Zimmermann (DE-Ilmenau)

9:00 a.m.	J. Scheufler (DE-Jena)
<p><b>MDD Tool Evaluation for Embedded Systems Within the Automation Domain</b></p> <p>The steadily increasing and hardly controllable complexity of modern software systems, as well as economic requirements like decreasing development-cycles within software development in general and the automation domain in particular raise the necessity of new engineering methods. The model driven development (MDD) approach promises to effectively handle these demands using platform-independent models of higher abstraction levels for software development, reusable transformation scripts and (automated) code generation out of these models. Within this paper the automation domain requirements onto MDD tools and their integration into a tool chain are presented. The features of modern MDD tools are correlated with the requirements of the automation domain. An interoperability matrix is presented, which enables the selection of UML-tools to be integrated into a MDD tool chain. Within a concrete case study a tool chain was set up, generating code starting with behavioral models.</p>	
9:20 a.m.	J. Werner (DE-Ilmenau)
<p><b>Concepts for a Model Driven Requirement Based Development Process</b></p> <p>The development of complex embedded systems usually is carried out by application of requirement based processes. Such processes are characterised by analysis and breakdown of textual requirements specifying the desired behaviour and other characteristics of the new system. This fact is also the main weakness because a plain textual description is not sufficient to obtain a complete and consistent image of a system. In opposition to that a completely model based process tries to describe the characteristics of a system within a formal model. The elaboration of such a formal model is carried out step by step whereas the first steps contain a high level of abstraction. In the later steps this abstraction is continuously replaced by the detailed formal description of the system. Implying an adequate modeling environment such a formal model can be used for simulation at any step of development. Thus validation and verification can be achieved much earlier. The main handicap of model based development lies in the increasing efforts in the early project stages wherein also a lot of experience of the involved developers is mandatory. This work tries to evolve concepts for a successful combination of both fundamental approaches. It shall be pointed out that restructuring requirement analysis by means of SysML models results in creating quality of specification. The application of UML models lower the risk of realising defective behaviour. Furthermore it will be shown that virtual integration in an executable cumulative model can be done for validating and verifying complex system behaviour.</p>	

9:40 a.m.	T. Jungebloud (DE-Ilmenau)
<p><b>Towards an Approach to Utilize Unified Modeling for Executable Specifications of Complex Systems</b></p> <p>Today's modeling languages like UML® (Unified Modeling Language) or SysML® (Systems Modeling Language) are well understood and frequently used by engineers of complex real-time- and embedded systems (RTES). Especially this domain has specific demands on the descriptive language used in the design process. Profiles for UML and SysML (e.g.: MARTE, UML Profile for Modeling and Analysis of Real-time and Embedded systems) tend to introduce facilities that convey aspects of scheduling, performance and time (SPT) in a standardized way. Nevertheless, a system specification in UML/SysML lacks of unified execution semantics. Such execution semantics are available in diverse domain specific simulation frameworks like MLDesigner. Moreover, especially the XML-based model description language MML (MLD Modeling Language) introduces a very similar concept to MARTE SPT called quantity-/server resources, which is mandatory for the modeling of functional and non-functional aspects of real-time- and embedded systems. In this contribution, we present an approach of modeling strategies and transformation rule sets to map UML models into MML models and vice versa. The transformation is explained in detail along with small example models to show the representation of model elements in the respective language.</p>	
10:00 a.m.	Ch. Bodenstern (DE-Ilmenau)
<p><b>Integrated software development of distributed automotive systems</b></p> <p>Over the last years methods of model based software development are widely established for design of complex systems. The design of distributed systems and the step from requirements towards first prototypes is much easier through using executable specifications. In modern automotive electrical and electronic systems a growing number of software functions have to be mapped on a given amount of electronic control units (ECU). Furthermore each ECU has very limited resources. Models based software engineering is the instrument for handling these issues. The need for a new design concept of automotive electronic systems and networks is shown by actual statistics of car break downs. In 2009 40% of all passenger car break downs were caused by electric or electronic issues (ADAC car break down statistics). Carmakers threat the lack of an adequate software design process by developing a new standard for automotive software. It`s called AUTOSAR. This standard has two main targets. First is the decoupling of hard- and software. The application is has no direct access to ECU hardware, so that it`s runnable on different ECU-types. The second innovation is a complete component based middleware which is one precondition for decoupling hard and software. The configuration of the AUTOSAR middleware is the most difficult task for carmakers changing their software development process to AUTOSAR. The main issue of the actual research project is the development of an integrated software design process using executable specifications to model the complete car within its environment, optimize the function distribution and generate the AUTOSAR middleware configuration together with the software components.</p>	
<b>10:20 – 10:40 a.m. Coffee break</b>	

10:40 a.m.	R. Schönefeld (DE-Ilmenau)
<p><b>Sequence-Based Specification as a New Kind of Modeling Embedded Systems</b>  Sequence-based specification (SbS) is a less known but powerful approach which formally specifies the software for an embedded or interactive system. The behavior of such a conceptual model of the software system (MSS) is expressible as a deterministic finite automaton with input and output (DFA/IO). The problem is to construct the state-transition function and the output function out of the functional requirements. In SbS both functions can be found by a so called enumeration schema of all possible sequences formed by all possible inputs crossing the interface(s) of the MSS and the required outputs. The paper will emphasize a new consideration of some details in the enumeration schema which support a more physical view on an MSS. It will be shown that there is an analogue between a mechanical system and a DFA/IO. That leads to interesting consequences for a software system concerning the relationship between a message (input) to the MSS and the state of the MSS and that the state carries the formal semantics of the MSS. Furthermore that for the semantics exist a "principle of least action" according to Hamilton's principle in mechanics. All these new insights offer better opportunities to find algorithms to reach a computer aided specification process.</p>	
11:00 a.m.	A. Karatkevich (P-Zielona Gora), S. Baranov (ISR-Ramat Gan)
<p><b>Graph Based Approach to Test Bench Constructing for Datapath</b>  Testing a datapath in a digital system such as a microprocessor or an embedded control system requires checking every possible way of sending data between the functional units at least once. Structure of connections between the units imposes some limitations on possibilities of such check. This paper considers a task of generating a test bench for given datapath, which covers every way of data sending at least once and is as short as possible. This paper presents a method in which a datapath is modeled by an oriented graph with nodes corresponding to functional units and arcs represent ways of data transfer. The task of optimal testbench generation can be then formulated as task of covering all arcs by the paths from input to output nodes of the graph with minimal total number of the arcs. Merging input and output nodes reduces the task to the Chinese Postman Problem, where the set of paths we seek can be obtained from the circuit being a solution of the CPP. The problem is complicated by the fact that the test sequences should consist of the microinstructions which may send data between several units at once and also by the fact that for different functional microoperations the units may require considering at different levels of abstraction. Modifications of the method taking it into account are considered in the paper. The proposed method is illustrated by a case study of testing a datapath of a processor unit.</p>	

11:20 a.m.	G. Łabiak (P-Zielona Gora)
<p><b>Some Computational Aspects of Statecharts Formal Modeling of Reactive Systems</b>  The paper presents a graphical notation for modeling of complex behavior of reactive systems. This notation, called statechart diagrams and invented by David Harel, features state-based description, concurrency and hierarchy. Other very essential characteristic of the diagrams is their very strict and formal definition, which allows to apply formal methods (e.g. based on state space characteristic function). This formal definition makes that, on one hand, statechart diagram can be directly implemented in programmable structure, and, on the other hand for example, their behavior can be analyzed against deadlock detection or can be transformed into other computational model (e.g. FSM). The paper concentrates on relations between some statechart semantic structures and their influence on the number of global states, what is very important for the computational complexity of formal methods algorithms.</p>	
11:40 a.m.	A. Bukowiec, A. Wegrzyn (P-Zielona Gora)
<p><b>From Modelling to Implementation of Concurrent Controllers by Means of Petri Nets, FSMs and Databases</b>  In the paper system for modeling and implementation of concurrent controllers is presented. Concurrent controllers are specified by Petri nets. Then structure of Petri nets is transformed into relational database. Based on data from database, Petri net is decomposed using symbolic method of analysis, like Thelen algorithm or coloring. Additionally the control algorithm can be checked if is well designed by checking if Petri net is bounded and live. In the result the set of finite state machines (FSMs) is received. Each FSM is implemented using methods of structural decomposition during process of logic synthesis. There is applied method of multiple encoding of microinstruction or internal states. It leads to decreased number of Boolean function realized by combinational part of FSM. The additional decoder could be implemented using embedded memory blocks. This leads to balanced usage of hardware resources of modern FPGA devices. The database is used for managing and changing the configuration of controller. It allows the remote reprogramming of whole control algorithm or only its part (one of concurrent FSMs).</p>	
12:00 noon – 1:30 p.m. Lunch	