07451 Abstracts Collection
Model-Based Engineering of Embedded Real-Time Systems
— Dagstuhl Seminar —

Holger Giese[1], Gabor Karsai[2], Edward Lee[3], Bernhard Rumpe[4] and Bernhard Schätz,[5]

[1] Univ. of Paderborn, DE
hg@uni-paderborn.de
[2] Vanderbilt Univ., US
gabor.karsai@vanderbilt.edu
[3] UC Berkeley, US
eal@eecs.berkeley.edu
[4] TU Braunschweig, DE
B.Rumpe@tu-bs.de
[5] TU München, DE

Abstract. From 04.11. to 09.11.2007, the Dagstuhl Seminar 07451 "Model-Based Engineering of Embedded Real-Time Systems" was held in the International Conference and Research Center (IBFI), Schloss Dagstuhl. During the seminar, several participants presented their current research, and ongoing work and open problems were discussed. Abstracts of the presentations given during the seminar as well as abstracts of seminar results and ideas are put together in this paper. The first section describes the seminar topics and goals in general. Links to extended abstracts or full papers are provided, if available.

Keywords. Models, model-based, MDD, embedded systems, real-time systems, validation & verification, tool-support, domain-specific languages

07451 Summary – Model-Based Engineering of Embedded Real-Time Systems

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.

Keywords: Models, model-based, MDD, embedded systems, real-time systems, validation & verification, tool-support, domain-specific languages

Dagstuhl Seminar Proceedings 07451
Model-Based Engineering of Embedded Real-Time Systems
http://drops.dagstuhl.de/opus/volltexte/2007/1271

Joint work of: Giese, Holger; Karsai, Gabor; Lee, Edward; Rumpe, Bernhard; Schätz, Bernhard

Extended Abstract: http://drops.dagstuhl.de/opus/volltexte/2007/1272

Model-Based Design Approach to Systems Integration

Ted Bapty (Vanderbilt University, USA)

A Model-Based Design Approach can have significant benefits when applied to the complex tasks of system integration. Succinct, formal, and complete representations of the target system can be used to:

- Ensure interface compliance
- Predict system properties, such as performance, security, safety, ...
- Generate software infrastructure
- Create system test harnesses
- Create system monitoring components
- Evaluate actual/implemented vs. simulated vs. specified performance

Uniting all of these capabilities together in one common represntation ensures consistency of simulations/analysis and implementations. We will discuss experiences in applying Model-Based design techniques to large-scale, service-oriented architecture systems, including modeling, analysis and generation.

Keywords: Model-Based, System Integration, Simulation

Model-Based Development of Fault-Tolerant Real-Time Systems

Christian Buckl (TU München, D)

Model-based development tools and code generators are frequently used at functional level. However in the context of safety-critical systems, the code generation of code at system level (communication in the distributed system, process management and scheduling, fault-tolerance mechanisms) would be even more important.

Two problems can be identified that need to be solved in this context. Current modelling languages like UML do not provide rigorous enough execution semantics for extensive code generation. In addition, the generated code is dependant on the used platform (hardware, operating system). Therefore, an extensible code generation framework has to be proposed.

In this talk, we will present solutions for both problems. We will present a domain specific model with explicit execution semantics (e.g. time-triggered computation model). In addition, we will present a model-based development tool based on templates for code generation.

Keywords: Fault-Tolerance, Code Generation, Model-Based Development

An Instrumentation-Based Approach to Controller Model Validation

Rance Cleaveland (Univ. of Maryland at College Park, USA)

This talk discusses the concept of Instrumentation-Based Validation (IBV): the use of model instrumentation and coverage-based testing to validate models of embedded control software. IBV proceeds as follows. An engineer first formalizes requirements as assertions, or small models, which may be thought of as monitors that observe the behavior of the controller model as it executes. The engineer then instruments the model with these assertions and develops test suites with the aim of highlighting where assertion violations occur. To make our discussion of IBV more concrete, we consider its implementation within the Reactis tool suite for the automated testing and validation of controller models given in Simulink / Stateflow.

Keywords: Model-based development, model-based verification and validation, monitors

Joint work of: Cleaveland, Rance; Sims, Steve; Smolka, Scott

Translation Validation of Generated Code through Equivalence Testing

Mirko Conrad (The MathWorks Inc. - Natick, USA)

Nowadays production code generation using Model-Based Design has replaced manual coding in various embedded application domains. Moreover, generated code is increasingly being deployed in safety-critical applications as well. To ensure quality and functional safety the embedded software has to be subjected to a variety of quality assurance measures.

In this talk I'd like to dicsuss a translation validation workflow for generated code which is based on equivalence testing between the executable model used as input for the code generation process and the object code resulting from the compilation of the code generator's output.

The proposed workflow, an engineering approach to translation validation, is inspired from practical experiences in the automotive industry.

Keywords: Model-Based Design, Simulink, code generation, translation validation, equivalence testing

TDL modeling in Ptolemy II

Patricia Derler (Universität Salzburg, A)

Embedded real-time systems consist of concurrent processes communicating with each other.

Processes can have different execution semantics; an important classification is into time-triggered and the event-triggered processes. Mixing those control paradigms in a heterogeneous system is difficult to model and simulate.

To research the behavior of time-triggered systems with event-triggered systems in heterogeneous models, we implemented a time-triggered model of computation, the timing definition language (TDL) in Ptolemy, a modeling and simulation environment for heterogeneous systems.

This talk presents the important concepts of TDL, the implementation of TDL in Ptolemy and shows how to build heterogeneous systems including TDL components in an example.

Keywords: Timing definition language (TDL), time-triggered, event-triggered, heterogeneous systems

Orthogonal Integration Techniques for AUTOSAR ECUs

Ulrich Freund (ETAS GmbH - Stuttgart, D)

Designing distributed embedded control systems is a complex task. Major parameters are not known in detail before a complete system has been built. Most prominent are control-algorithm tuning and real-time analysis.

In control-algorithm design using abstract controller models which are tested against models of a plant is the very first step in system design. These models typically do not consider the computational limitations of a microcontroller and use floating point arithmetics. If these controller models run well in the vehicle model, they are transferred to a rapid-prototyping system driving sensors and actuators. Thus the model of the control-algorithms runs in a real car. At this point, the controller models are executed in real-time. A typical next step is to transform the controler model to fix-point arithmetic and validate the results in a real-vehicle. Since in a typical closed-loop diagram representation the controller is left to the plant, but on one pane, we call the control-algorithm validation horizontal integration.

AUTOSAR focus on the exchangability of software-components between different ECUs. To achieve this goal, AUTOSAR uses a software component description, a runtime-environment (RTE) as middleware and a lot of configurable basic software modules. The execution semantics of the RTE poses constraints on the partioning of the control-algorithms to so-called runnable-entities while the data-elements in interfaces prescribe the resolution of the signals, typically assuming fixed point arithmetic.

Configuring a running AUTOSAR ECU with its RTE and basic-software modules is a complex task in its own. Because the AUTOSAR ECU software architecure defines several layers, we call the successfull configuration of an AUTOSAR ECU, validated against test signals, e.g. CAN, vertical integration.

Evaluation projects have shown that adapting vehicle-validated fixed-point control-models can easily mapped to AUTOSAR software components and cru-

cial configuration parameters like signal-to-frame- and runnable-entity-to-task-mapping can be derived from the horizontal integration, thus limiting the vertical integration effort considerably.

At Dagstuhl, we will present a combined horizontal/vertical, i.e. orthogonal integration of an engine-management-system at an early stage of design.

Keywords: AUTOSAR, Integration, Validation, Closed-Loop-Control, Model-based-design

Joint work of: Freund, Ulrich; Frey, Patrick

Abstraction and the Model-Based Engineering of Advanced Softwareintensive Systems*

Holger Giese (Universität Paderborn, D)

A general argument often stated is that model-based engineering can be employed to raise the level of abstraction and thus in turn allows us to better deal with problems of higher complexity. This talk will at first discuss that two major kinds of abstractions are employed in our field. We will then review which possibilities we have to support the model-based engineering and how they are related to these two different forms of abstraction. In addition, the relation between abstraction and decomposition will be discussed and we will present some examples for compositional analysis opportunities which have been developed using specific abstraction and decomposition concepts for softwareintensive systems with advanced capabilities such as multiple variants or online-reconfiguration.

*This work was developed in the course of the Special Research Initiative 614 – Self-optimizing Concepts and Structures in Mechanical Engineering – University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft.

Development of specifications for modular verification

Susanne Graf (VERIMAG - Gières, F)

The SPEEDS project proposes a meta-model HRC for improving the development of embedded systems (or systems of systems). It is intended to be expressible as a SysML profile and proposes mainly 3 novelties with respect to SysML:

- extension of components with contracts
- representation of heterogenous systems (including hybrid and restricted probabilistic behaviours)
- a formal composition semantics,
- a richer set of connectors (composition operators).

A set of analysis tools will be connected allowing to validate correctness of contract hierarchies and satisfaction of properties by abstract designs. A system advisory tool will monitor and drive the overall progress.

Here, we will discuss the problem of defining useful contracts (expressing safety properties). We will show how to exploit "usecases" for this purpose and illustrate on a case study. If there is time, we will show how to extend this approach to progress properties.

Obversations on Live Elements in Behavioral Specifications

Michaela Huhn (TU Braunschweig, D)

Since the new major version 2 of the UML, the suitability of activity diagrams for modeling requirements has increased significantly. UML 2 activity diagrams are based upon a completely reengineered metamodel including many new features and an improved semantic precision. We provide a formal syntax and operational semantics for activity diagrams to allow for fully executable models. Inspired by the scenario-based language of Live Sequence Charts (LSC), some extensions for activity diagrams are proposed including a distinction between possible and mandatory behavior. The proposed semantics paves the way for formal reasoning and tool development that allows for early prototyping and validation by simulation.

Moreover, we report on experiences with synthezising statecharts from Live Sequence Charts specifications.

Keywords: UML Activity Diagrams, Formal Semantics

Integrated Modeling for Safe Transportation

Hardi Hungar (OFFIS - Oldenburg, D)

Traditional model-based design establishes models of the embedded system and of relevant parts of its technical environment and exploits these for an analysis of their joint dynamics.

Due to the current evolution of embedded functionality from embedded control to operator assistance systems, the overall dynamics is gradually moving towards a human-in-the-loop behaviour. As the dynamics of such systems cannot be fully understood without co-modelling and co-analysis of all the agents involved, including the human operator, the project IMoST ("Integrated Modelling for Safe Transportation") addresses a seamless semantic integration of the most appropriate modelling paradigms for those individual agent types. This entails detailed models of human operator behaviour - both normative and erratic - in road traffic situations as well as models formalizing multiple viewpoints of embedded system dynamics, and models of the environment as perceived and partially controlled by the embedded system and the human operator. The

project's goal is to be able to demonstrate the viability of this approach, by deriving an instance of such a model for a particular task in road traffic, and appropriate analysis techniques, both validated by extensive experiments in a realistic.setting.

This talks provides an overview of the project and its current state. IMoST is funded by the Ministry of Science and Culture of the state of Lower Saxonia.

Keywords: Model-based development, human-in-the-loop, safe transportation, automotive

Modeling Models of Computation

Gabor Karsai (Vanderbilt University, USA)

Models of Computation (MoC-s) play a fundamental role in model-driven development: they define what a component is and how components could interact. Their precise definition is essential, yet few results are available that about their formal specification in an operational, yet analyzable form. The work described here shows some early results about how this can be done in a timed, discrete-event framework. The talk will present the approach and a few examples for the actual specification of well-known MoC-s.

A Service-Oriented View on Failure Management

Ingolf Krueger (University of California, San Diego, USA)

Failure management for complex, embedded systems-of-systems is a challenging task. In this presentation we argue that an interaction-based service notion allows capturing of failures end-to-end from both the development-process and the runtime perspective. To that end, we develop a formal system model introducing services and components as partial and total behavior relations, respectively. A failure-ontology, refined for the logical and the deployment architecture is presented. This ontology allows us to reduce the partiality of services by managing failures.

Furthermore, the ontology gives rise to an architecture definition language capturing both logical and deployment architecture models together with a failure hypothesis. Finally, we exploit these models for the formal verification (model-checking) of properties under the given failure hypothesis.

Keywords: Service, Service-Oriented, Failure Management, Model-Based Development, Embedded Systems, Systems-of-Systems, Formal Verification

Heterogeneous Modeling in the Ptolemy Project

Edward Lee (Univ. California - Berkeley, USA)

The Ptolemy Project studies modeling, simulation, and design of concurrent, real-time, and embedded systems. The focus is on assembly of concurrent components whose interactions are governed by well-defined models of computation. The MoCs include process networks, rendezvous-based models, dataflow, discrete-event models, synchronous/reactive models, and finite state machines. Heterogeneous combinations of these MoCs yield modeling techniques such as Statecharts and hybrid systems. This talk focuses on mixtures of these models of computation and related software infrastructure with particular attention to possibilities for distributed real-time computing, and with attention to relationships with currently popular modeling techniques such as UML.

Keywords: Models of computation, concurrency, real-time systems, distributed systems, heterogeneity

Full Paper:
http://ptolemy.eecs.berkeley.edu

Draft Abstraction Map

Edward Lee (Univ. California - Berkeley, USA)

These slides, which I showed at the Tuesday panel, represent a draft of a partial order among models in a system design, based on the notion of abstraction. They illustrate that integration typically today happens at the greatest lower bound of the models of interest. Our goal should be to perform integration instead at the least upper bound.

Keywords: Abstraction, multi-view models, integration

Virtual Prototyping and Model-Based Engineering with the Model of Components "42"

Florence Maraninchi (VERIMAG - Univ. of Grenoble, F)

Model-Based Engineering is about using models of various abstraction levels along the flow from the initial design to the actual implementation. The flow may be entirely automated or not, but at least there is a requirement for consistency between the various levels of abstraction along the flow.

Virtual Prototyping is about using a sufficiently detailed model for intensive simulations, before starting the actual development, for instace in order to choose the value of some critical parameters. It requires executable models, but

not necessarily a chain of models from the very first simulation to the actual implementations.

Of course Virtual prototyping+model-based engineering is a desirable goal. We report on some experiences in Virtual Prototyping (VP), or Model-Based Engineering (MBE), and we explain what it means to do both for the same system.

Then we present a component-model called 42, which could serve in making VP and MBE usable consistently in the same project.

A first version of 42 has been presented at GPCE'07. Here is a brief description:

Every notion of a component for the development of embedded systems has to take heterogeneity into account: components may be hardware or software or OS, synchronous or asynchronous, deterministic or not, detailed w.r.t. time or not, detailed w.r.t. data or not, etc. A lot of approaches, following Ptolemy, propose to define several "Models of Computation and Communication" (MoCCs) to deal with heterogeneity, and a framework in which they can be combined hierarchically. 42 aims at expressing fine-grain timing aspects and several types of concurrency as MoCCs, but we require that all the MoCCs be "programmed" in terms of more basic primitives. 42 is meant to be an abstract description level, and a reasoning tool for understanding what it means to use VP and MBE together, in particular for embedded systems.

Keywords: Virtual Prototyping, Model-Based Engineering, system-level modelling, components, 42

Joint work of: Maraninchi, Florence; Bouhadiba, Tayeb

See also: 42: Programmable Models of Computation for a Component-Based Approach to Heterogeneous Embedded Systems F. Maraninchi, T. Bouhadiba. Sixth ACM International Conference on Generative Programming and Component Engineering (GPCE'07), October 1-3, 2007, Salzburg, Austria

Model-Based Design of Embedded Systems

Pieter J. Mosterman (The MathWorks Inc. - Natick, USA)

Model-Based Design is presented as consisting of four elements: (i) executable specification, (ii) design with simulation, (iii) implementation through code generation, and (iv) continuous test and verification. This presentation concentrates on the combined design and implementation as model elaboration. It is shown how the design of an edge detection filter can be systematically brought to an implementation by comparing a reference algorithm to an increasingly detailed representation of the implementation. Automatic program synthesis allows the generation of C code or a representation in a hardware description language (HDL). The HDL emulation can then be co-simulated in the system context to study behavior of an implementation at a cycle accurate level. This reduces

expensive hardware iteration, facilitates analysis of system characteristics with detailed component implementation models, and mitigates the need for extensive testbench design.

Design Space Exploration in Model Based Design

Sandeep Neema (Vanderbilt University, USA)

A dominant aspect of model-based design deals with structural representation of system designs. The conventional approach to model-based design relies on strictly deterministic structural representation with hierarchical decomposition. Arguably, determinism has it merits, however, in large and evolving systems design the deterministic representation often forces suboptimal decisions without sufficient context. In this presentation we describe domain independent constructs with which structurally deterministic domain specific languages can be augmented to incorporate non-determinism. We call such augmented representation - design space representations and describe domain independent tools with which design spaces could be systematically manipulated. We will show the application of these tools and concepts in a variety of domains.

Keywords: Design Space Exploration

The Carolo Project – Impressions from the Darpa Urban Challenge 2007

Bernhard Rumpe (TU Braunschweig, D)

The Darpa Urban Challenge had its finisch on Nov. 3rd. Initially roughly 100 groups participated in the race, where a car had to autonomously drive 60 miles in urban environment. 4wayy stop, precedence resolution, surpassing obstacles, free parking, etc. were among the duties of the car.

The Braunschweig team made a good 7th place.

This are the impressions from the semifinals and finals accompanied with a discussion, where model based development was of help and where it wasn't.

Keywords: Darpa Urban Challenge

Joint work of: Rumpe, Bernhard; Thomas Form, Peter Hecker, Marcus Magnor, Walter Schumacher, Lars Wolf

Full Paper:
http://carolo.tu-bs.de/

A test model quality framework

Ina Schieferdecker (TU Berlin, D)

Test effectiveness is a central quality aspect of a test specification which reflects its ability to demonstrate system quality levels and to discover system faults.

A well-known approach for its estimatation is to determine coverage metrics for the system code or system model. However, often these are not available as such but the system interface only, which basically define structural aspects of the stimuli and responses to the system.

Therefore, this presentation focusses on the idea of using test data variance analysis as another analytical approach to determine test quality. It presents a method for the quantitative evaluation of structural and semantical variance of test data. Test variance is defined as the test data distribution over the system interface data domain. It is expected that the more the test data varies, the better the system is tested by a given test suite. The presentation instantiates this method for black-box test specifications written in TTCN-3 and the structural analysis of send templates. Distance metrics and similarity relations are used to determine the data variance.

While the structural test data analysis for functional tests has been presented at TestCom 2007, this presentation will add first ideas on how to expand the approach into Continuous TTCN-3 test specifications for embedded systems. The discussion is put into a more general framework of aspects of test model quality.

Keywords: Model quality, test effectiveness, test data variance, test optimization

Clone-Detection in Model-Based Development of Embedded Control Functions

Bernhard Schätz (TU München, D)

Model-based development is becoming an increasingly common development methodology. In important domains like embedded systems already major parts of the code are generated from models specified with domain-specific modelling languages. Hence, such models are nowadays an integral part of the software development and maintenance process and therefore have a major economic and strategic value for the software-developing organisations. Nevertheless almost no work was done on a quality defect that is known to seriously hamper maintenance productivity in classic code-based development:

Cloning. This talk presents an approach for the automatic detection of clones in large models as they are used in model-based development of control systems. The approach is based on graph theory and hence can be applied to most graphical data-flow languages. An industrial case study demonstrates the applicability of our approach for the detection of clones in Matlab/Simulink models that are widely used in model-based development of embedded systems in the automotive domain.

Joint work of: Schätz, Bernhard; DeiSSenböck, Florian; Hummel, Benjamin; Jürgens, Elmar; Wagner, Stefan

Model-Based Generators for Domain-Specific Simulations

Jonathan Sprinkle (Univ. of Arizona - Tucson, USA)

This research bridges two main uses of model-based design: reachability analysis (in either continuous or discrete states), and simulation. The goal of the work is driven by human interaction with UAVs, especially multi-vehicle simulations with mixed-participation by some humans and some autonomy. By considering the continuous reachability of the vehicle, with the discrete reachability of the state machine that describes (a formalization of) the interaction protocol, we can identify initial conditions which may serve as points of interest for simulation with a human in the loop. Further, we can begin to formalize the decision authority model, which describes in which cases an autonomous vehicle might be permitted to disobey a message which may have been delayed by network or human latency.

Joint work of: Ding, Jerry; Tomlin, S. Claire; Sastry, Shankar

Keywords: Embedded Humans, Reachability, Code Generation

Relating Computer Systems to Sequence Diagrams

Ketil Stølen (SINTEF - Oslo, N)

Having a sequence diagram specification and a computer system, we need to answer the question: Is the system compliant with the sequence diagram specification in the desired way? In this paper we present a procedure for answering this question for sequence diagrams with underspecification and inherent nondeterminism. The procedure is independent of the implementation technologies used for the system, and relies only on the execution traces that may be performed by the system. Similar to testing, our procedure may result in one of three possible verdicts: "compliant", "not compliant" and "don't know". The last verdict, "don't know" is a result of the fact that we may know only a subset of the system traces. If all system traces are known, the procedure results in either "compliant" or "not compliant". We also characterize situations where a definitive verdict may be given even in the case where only a subset of the system traces is known.

Keywords: Sequence diagrams, refinement, compliance, trace semantics

Joint work of: Refsdal, Atle; Runde, Ragnhild Kobro; Stølen, Ketil

Explicit Modeling of Semantics for Domain Specific Modeling Languages

Janos Sztipanovits (Vanderbilt University, USA)

Domain-Specific Modeling Languages (DSMLs) play fundamental role in the model-based design of embedded software and systems. While abstract syntax metamodeling enables the rapid and inexpensive development of DSMLs, the specification of their semantics is still a hard problem. This presentation will discuss methods for the explicit modeling of structural and behavioral semantics of DSMLs. The discussion will be based on our previous work on methods and tools for the semantic anchoring of DSML-s. Semantic anchoring translates DSMLs into carefully selected mathematical domains: term algebra and logic for structural semantics and Abstract State Machines for behavioral semantics. In this presentation, we show use cases for structural and behavioral semantics and discuss new research on extending the semantic anchoring framework to heterogeneous behaviors by means of composing semantic units.

Keywords: Domain Specific Modeling Languages, Semantics, Semantic anchoring

Three Problems of Model-based Design

Janos Sztipanovits (Vanderbilt University, USA)

The presentation discusses three common problems in model-based design: (1) Misunderstanding the concept of orthogonality across design concerns, (2) Missing the fundamental importance of structural semantics of DSMLs, and (3) Not recognizing the need for the explicit representaiton of behavioral semantics of DSML-s.

Keywords: Model integrated computing, domain specific modeling languages, structural semantics, behavioral semantics

Towards a framework and methodology for model based engineering of embedded real-time systems

Martin Törngren (KTH - Stockholm, S)

A framework for model based engineering (MBE) ought to answer at least the questions Why, When, What and How MBE?

We start out by briefly describing alternatives to MBE, lessons learnt from mature engineering disciplines, as well as highlights from problems experienced with MBE today in the area of embedded systems.

As a basis for the framework it is necessary to provide a contextual perspective that relates the role of MBE to product aspects, technical and management processes, and organizational roles. Such a perspective also enables to identify requirements on technology, and potential mismatches between MBE technology and its context. Our conceptual framework allows to reason about MBE drivers (i.e. when and why to introduce MBE), explains the various dimensions of MBE and provides guidelines for adopting MBE (how). The framework supports qualitative reasoning about the potential benefits of applying MBE, however quantification of such benefits remains a challenge.

We then turn to discuss concrete instances of the framework for embedded real-time systems and in particular highlight

- Connections between model management and views, and between requirements, design and V&V information
- Relations between platforms, functions and system qualities, emerging vs. designed properties.
- Challenges for dynamically configurable systems.

Examples and experiences will where appropriate, and given the time, be taken from the following projects: ATESST (www.atesst.org), DYSCAS (www.dyscas.org) and other projects including AIDA.

Joint work of: Törngren, Martin; Chen, DeJiu