Model-Based Design of Automotive RT Applications
Modeling approach

- Modeling concept
- Realization in tool chain
- Use cases

Challenges in the automotive environment

- The automotive electronics challenge
- Some answers
Part I

Modeling approach
Modeling Concept

Architectural components and their relations

- Tasks execute algorithmic functions and logic functions
- Tasks transmit and receive signals (local and global)
- Tasks are simple tasks (no synchronisation during execution)
- Tasks run on hosts
- Hosts contain one or more controllers
- Controllers transmit and receive signals
- Controllers are connected to a network
- Controllers on one network form a cluster
**Modeling Concept**

**Synchronous system model**
- Known and bound maximum execution time of tasks
- Known and bound maximum transmission duration of signals
- Known minimum interarrival time of task activation triggers
- Known minimum interarrival time of transmission requests for signals

**Periodic and phase locked system model**
- An application cycle defines the basic periodicity of the application
- Task execution is triggered periodically, period is an integral divider of application cycle
- Task execution is triggered with a defined time offset to start of application cycle
- Signal transmission is triggered periodically, same rule for period
Modeling Concept

- Requirements
- Functional Model
- 'A'-Model
- Source Code
- Test
- Validation
- Verification

DECOMSYS
Functional Model

Model of application function

- Control algorithms, state machines, logic algorithms
- Model of environment
- Segmentation of model in particular functional parts
- Model does not contain architectural information

Benefits

- Check basic correctness of application algorithms
- Full access to all system signals for testing and debugging
Functional Model

<table>
<thead>
<tr>
<th>Objects</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control loop</td>
<td>Control period</td>
</tr>
</tbody>
</table>

A/D
Architecture-Allocated Functional Model

Assignment of model parts to architectural components

- Assignment to hardware (host, network)
- Assignment to software (task)
- Communication modelled by generic connectors

Benefit

- Defines the hardware and software architecture for the application
- Allows generic simulation of task activation and communication
Architecture-Allocated Functional Model

Objects

<table>
<thead>
<tr>
<th>Control loops</th>
<th>Task periods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hosts</td>
<td>Task offsets</td>
</tr>
<tr>
<td>Tasks</td>
<td></td>
</tr>
<tr>
<td>Generic connectors</td>
<td></td>
</tr>
</tbody>
</table>

Parameters

- Task periods
- Task offsets

ECU 1

ECU 2

ECU 3
Architecture Model

Pure architecture information

- Software model (tasks and signals forming transactions)
- Hardware model (hosts, networks)
- Relevant parameters
  - Task timing (period, offset, WCET)
  - Task communication relations (signals transmitted and received)
  - Signal parameter (period, offset, signal representation)
  - Assignments of tasks to hardware components

Benefits

- Mapping of transactions to a communication technology
- Generation of configuration data for OS, middleware and communication controllers
**Objects** | **Parameters**
---|---
Hosts | Task offsets
Tasks | Signal periods
Generic connectors | Signal offsets
Network topology | Message schedule
Virtual Prototype

Extension of the architecture-allocated functional model

- Generated technology mapping parameters are added to the AAFM
- Generic simulation modules are exchanged by specific technology simulators

Benefit

- Simulation of application function under influence of operating system and communication
  - Task activation and preemption simulation
  - Communication timing and scaling simulation
  - Startup and restart simulation
  - Full access to all system signals for testing and debugging
- Model based fault-injection
- Model based testing
Realization in tool chain

- Functional Model
- Architecture Allocated Functional Model (AAFM)
- Architecture Model (AM)
- Middleware Code (MW)
- Virtual Prototype (VP)
- SIMSYSTEM
- SIMCOM
- SIMTARGET
- Source Code

DESIGNER
GENERATOR

SIMSYSTEM
Technologies

Modeling tool
- MATLAB/Simulink
- DECOMSYS blocksets

Code Generation
- Real-Time Workshop
- Embedded Coder in preparation

Technology Mapping and software environment
- DECOMSYS tools and OSEKtime/FTCom for FlexRay communication protocol
- 3SOFT ProOSEKtime/OS
**SIMSYSTEM**

### Architectural design elements
- Hardware: networks, hosts
- Software: tasks, connectors

### Function
- Blockset library
- Simulation of time-driven task activation
- Generic simulation of communication without technology information
- Automatic generation of and interaction with architectural model
SIMSYSTEM Screen Shots
**Communication technology simulation**

- Technology specific simulation core
- Uses configuration parameters from architectural model

**Function**

- Simulates signal transmission timing
- Simulates signal scaling effects
- Simulates middleware influence (FTCom)
- Simulates OS synchronous to communication system
- Simulation based fault injection
Use Cases

Application Development for single electronic control unit

Model Based Testing
Application Development for Single ECU

**Development of functional model**
- Application functions
- Environment simulation functions
- Stimuli generation functions

**Architecture specification**
- Separation of ECU functions and remaining system functions
- Definition of software and hardware architecture

**Technology mapping**
- Scheduling for communication technology
- Generation of configuration parameters
Application Development for Single ECU

**Virtual Prototype**
- Testing and validation of application functions in simulation

**Application to real hardware**
- Code generation or hand coding for ECU
- Code generation for remaining system separately
- Application of remaining system code to power node
- Connect single ECU and power node via bus system
- Testing of ECU function against model running in power node
Development of functional model

- Same as for application development

Assertion functions

- Additional assertion functions are added to functional model
- Assertion functions define correctness conditions that must hold for application
- Assertion functions use signals as input and generate boolean statements

Execution

- Assertion functions are active in virtual prototype by simulation
- Assertion functions are executed in real hardware by generated assertion code on power node
Part II

Challenges in the automotive environment
Automotive Electronics Challenge

Challenges for future electronic architectures

- Reusability of software functions
- Standardization of software infrastructure
- Scalability of subsystems for multiple model classes
- Dr. Thomas Scharnhorst, VW EES at 3SOFT Automotive Day 2003

Other requirements

- Support for collaborative development
- Integration in existing processes and tool environments
- Integration with other tools along the V-model

Design and modeling tools should support these requirements
Key architectural properties for tool chain

- Separation between
  - functional model
  - software model
  - hardware model and technology

- Mechanisms to
  - Import and export functional model parts
  - Extract and apply software and hardware architecture information
  - Import and apply technology mapping parameters

- Open and standardized interfaces
  - For integration in existing tool environments
  - For OS and communication middleware (for code generation)
Use of model-based design tools in automotive industry

- Still not commonly used
- Still development to do
- Interest in processes for development of safety critical application will increase interest in model-based design tools
- Economic pressure and complexity are other motivators towards model-based design approaches