Modeling and Simulation of Distributed Automotive Systems with Simulink®

MathWorks IAC 2006
Stuttgart
May 16th 2006
Dr. Thomas Ringler
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>State of the Art</td>
</tr>
<tr>
<td>Challenges for Modeling Distributed Systems</td>
</tr>
<tr>
<td>SimEvents Case-Study</td>
</tr>
<tr>
<td>Conclusions and Outlook</td>
</tr>
</tbody>
</table>
State of the Art

- Model-Based Design and code-generation has reached series development within DaimlerChrysler

- Model-Based Design is used amongst others for central ECUs in the body domain

- Characteristics of the body domain
  - High distribution of functions on a high number of ECUs
  - Concentration of many functions at the central ECUs
  - “Soft” real-time requirements

- Suppliers are part of the model-based workflow
State of the Art
Current model-based Workflow

- Functional executable specifications
- Represented in independent Simulink® models
- Validation of functions

- Result: Functions integrated in ECU
- Validation of interaction of functions

Goal: Virtual validation of the interaction of ECU-functions and distributed functions
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>- State of the Art</td>
</tr>
<tr>
<td>- Challenges for Modeling Distributed Systems</td>
</tr>
<tr>
<td>- SimEvents Case-Study</td>
</tr>
<tr>
<td>- Conclusions and Outlook</td>
</tr>
</tbody>
</table>
Challenges for Modeling Distributed Systems

Current State:
Modeling independent functions

Use-Case 1:
Modeling function-compound located on an ECU

Use-Case 2:
Modeling distributed functions (e.g. Light Control)

Functional Aspects
- Validation of interfaces and functional behavior
  - Init. and shutdown
- Validation of interrelationship of ECU-functions
  - ECU related Power-Up/-Down
- Validation of interrelationship of distributed functions
  - Global vehicle states

Temporal Aspects
- Abstracted execution models
- Temporal interrelationship of functions and basic software
- Temporal behavior of communication systems

State of the art
Additional requirements towards modeling-tools
Challenges

Tool support for structuring models

- Encapsulation of different functions in one large model
- Enhanced usability and conciseness for large models
  - Concepts for reducing wiring effort
  - Configurable complexity regarding interfaces and subsystems
- Separation of functions from the execution control, ECU-services and communication
- Modeling basic features of ECU-services and communication
## Challenges

### Tool support for Network Communication Modeling

- **Network communication modeling**
  - Objective: validate model interrelationship of distributed functions
  - Regard only to relevant temporal behavior of basic software and communication

- **Detailed CAN-arbitration modeling not needed**
  - Simple delay model sufficient
  - Configuration for typical, best-case, worst-case delays
  - Support for easy tunable delays and statistical distribution of delays needed

- **Simulation of communication within Simulink®**
  - Export/tool-coupling with communication-simulators is not very practical
Contents

- State of the Art
- Challenges for Modeling Distributed Systems
- SimEvents Case-Study
- Conclusions and Outlook
SimEvents Overview

- Allows modeling discrete-event systems within Simulink®
- Time-based and discrete-event-based modeling within one Simulink® model
- Blocks known from queuing theory
  - Generation of entities
  - Gates for controlling the entity flow
  - Servers for modeling temporal behavior of entities
  - Queuing entities
  - Assignment of attributes to entities

- Aims to be especially suited for modeling networks and distributed systems
SimEvents Case-Study

- **Objective**
  - Understand basic concepts of structuring
  - Transfer basic concepts of distributed automotive systems to SimEvents

**Diagram:**
- Node 1:
  - Appl. → Fcn
  - Com. Layer: Rx → Tx
- Node 2:
  - Appl. → Fcn
  - Com. Layer: Rx → Tx

**Communication Network CAN**
SimEvents Case-Study
Top-Level

Entities passing messages between nodes and network
SimEvents Case-Study
Node Modeling

- Entity containing received signals
- Application task
- Entity containing application signals
- Communication transmit task
- Communication receive task
- Entity containing CAN message
- Connector from network
- Connector to network

Application tasks
Communication tasks
Communication Network CAN
SimEvents Case-Study
Application Task Modeling

- Application task generates output signals based on received signals
SimEvents Case-Study
Communication Layer / Transmit Task Modeling

- Transmit task sends CAN-message based on application signals
- Oversampling

Signals are extracted from the entity
Signals are passed as entity-attributes
Communication related attributes are assigned to the entities
Connector to network
Oversampling: New entities with higher activation rate are generated
SimEvents Case-Study
CAN Communication Delay Modeling

- Network delays CAN-message

Paths from different nodes are combined

Delay of entities by an infinite server
Individual delay times have been assigned to each CAN message before

Entities are replicated for the different receivers (broadcast)

Scope for entity attributes
SimEvents Case-Study
Communication Layer / Receive Task Modeling

- Receive task filters CAN-message and provides signals for the application
- Overwrite semantics of receive buffer

Message receive filter based on CAN-ID

Accepted entities pass the switch

Entities pass the gate when task is triggered

Connector from network

CAN-ID attribute is extracted from entity

Receive buffer as single server

Overwrite semantics of receive buffer, latest message wins

Connector to application
SimEvents Case-Study
Simulation Results

Signal-latency

Signal sent

Signal received

Less samples received than sent

CAN-Messages on Network
Contents

- State of the Art
- Challenges for Modeling Distributed Systems
- SimEvents Case-Study
- Conclusions and Outlook
Conclusions and Outlook

- First result: Concepts of distributed automotive systems could be modeled
- SimEvents provides a powerful and scalable approach
- Suited for overall temporal aspects of distributed systems
- Big advantage: modeling functional behavior and communication in one tool
- Different paradigm: “Thinking in entities”
- Additional features could help to make modeling more convenient
  - Support of Simulink® busses as attributes of entities
  - Configuration based on CAN-Communication description files

▶ SimEvents is another building block for virtual validation of distributed systems
▶ For productive use, customized solutions based on CAN-configuration needed

Outlook

- Derive modeling patterns, provide reusable blocks
- Virtual validation of exemplary series-models