VIRTUAL SYSTEM PROTOTYPING IN AUTOMOTIVE INDUSTRY AND THE ROLE OF FUNCTIONAL MOCK-UP INTERFACE.

7TH MODPROD WORKSHOP ON MODEL-BASED PRODUCT DEVELOPMENT.
AGENDA.

1. Function Development in Automotive Industry
2. Integration of Functions
3. Improve Methods: Virtual System Prototyping
4. Role of Functional Mock-Up Interface
5. Example Formula Student Electric
AGENDA.

1. Function Development in Automotive Industry
2. Integration of Functions
3. Improve Methods: Virtual System Prototyping
4. Role of Functional Mock-Up Interface
5. Example Formula Student Electric
ICONIC CHANGE.
BMW MOBILITY IN CHANGE.

yesterday

![Yesterday Image](image1)

![Yesterday Image](image2)

![Yesterday Image](image3)

today

![Today Image](image4)

![Today Image](image5)

 tomorrow

![Tomorrow Image](image6)

![Tomorrow Image](image7)

![Tomorrow Image](image8)
BMW DRIVER ASSISTANCE SYSTEMS. CAR IMPLEMENTS FUNCTIONALITY.
EXAMPLE INTELLIGENT LIGHT.
AGENDA.

1. Function Development in Automotive Industry
2. Integration of Functions
3. Improve Methods: Virtual System Prototyping
4. Role of Functional Mock-Up Interface
5. Example Formula Student Electric
BMW DRIVER ASSISTANCE SYSTEMS.
MORE COMFORT, MORE SOUVERANITY, MORE SAFETY.
REQUIREMENTS TO INTEGRATED FUNCTIONS.

Integrated Functions

- Weight
- Ergonomic
- Drivability
- Styling
- Quality
- Environment / Emission
- Cost of Ownership
- Ride Comfort
- Handling
- Safety
How can we manage?
INTEGRATION OF MULTI DOMAIN SYSTEMS AND COMPONENTS.
INTEGRATION OF SIMULINK, CRUISE, DYMOLA, C-CODE.
How can we manage?
WE NEED EVOLUTION JUMP WITH NEW METHODS AND TOOLS.
How can we manage?
AGENDA.

1. Function Development in Automotive Industry
2. Integration of Functions
3. Improve Methods: Virtual System Prototyping
4. Role of Functional Mock-Up Interface
5. Example Formula Student Electric
HOW CAN WE MANAGE?

Virtual Rapid Prototyping:
Combine Virtual Integration,
Real Test Experience and
Design Space Exploration
CLASSICAL DEVELOPMENT APPROACH.

But, how we know if we have done the right things?
HOW TO ACHIEVE AN EARLIER VEHICLE EVALUATION & VALIDATION?

Approximately 60% of development time no real prototypes are available!

Source: Customer survey
HOW TO ACHIEVE AN EARLIER VEHICLE EVALUATION & VALIDATION?

Less than 10% of the engineers get evaluation experience in the global vehicle.
VIRTUAL TEST DRIVING IS BASED ON REAL LIFE TEST SCENARIOS.
EXAMPLE VISUALIZATION.
VIRTUAL TEST DRIVING IS BASED ON REAL-WORLD EXPERIENCE.
EXAMPLE ANIMATED PEDESTRIAN.
REPRODUCE THE COMPLEX REAL TEST DRIVER JOB.

Full vehicle control

State observer

Operation of test device
• Data Acquisition Device
• Failure Insertion
• Diagnostic
• Calibration Tools
• Steering & Braking Machine
EARLY VALIDATION OF SYSTEMS IN THE WHOLE VEHICLE.
WORKING TOGETHER FOR THE FIRST TIME RIGHT.

Requirements and Performance Targets

Virtual System Design: Modeling & Design Space Architecting

Early Validation & Decision

Virtual System Integration: Simulation & Design Space Evaluation

Virtual Vehicle Integration
Virtual Prototyping allows to...

a) identify the best solution amongst others
b) validate it against requirements
c) reduce risk & cost
Virtual Prototyping consists of three major process steps
1) Building up the design space  
2) Modeling solutions  
3) Identifying the best solution
HOW CAN WE MANAGE?

Virtual Rapid Prototyping:

Combine Virtual Integration, Real Test Experience and Design Space Exploration
HOW CAN WE MANAGE?

System Simulation:

Combine Aspects of Plants and Controllers
COMBINE PLANT AND CONTROLLER SIMULATION: SYSTEM MODELING, INTEGRATION, AND SIMULATION.
Solution

Function Development

Integration

Debug

Validation / Test

ASCET
MATLAB / SIMULINK
ISOLAR-EVE
ISOLAR-A
AUTOSAR Builder
ECU-TEST
RT2

Public | ETAS-PGA/EAP | 01/14/2013 | © ETAS GmbH 2012. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
EXAMPLE: DEBUGGING SOFTWARE ERROR.

```java
/* Logic: 'S4'/Logical Operator' incorporates:
 * Import: 'S4'/ACC.Enabled
 * Import: 'S4'/Target.Detected'
 */

rtb_targetdetected = ((AccelCtrl_ACC2_U.ACCEnabled != 1.0) &&
                      (AccelCtrl_ACC2_U.TargetDetected != 0.0));

/* Switch: 'S4'/Switch1' incorporates:
 * Constant: 'S4'/Target dv
 * Import: 'S4'/Target.NearPoint_dv
 */

if (rtb_targetdetected) {
    rtb_dvms = AccelCtrl_ACC2_U.TargetNearPoint_dv;
} else {
    rtb_dvms = AccelCtrl_ACC2_P.Targetdv_Value;
}
```
INCREMENTAL SYSTEM IMPROVEMENT.

```c
/* Logic: '<S4>/Logical Operator' incorporates:
 * Inport: '<Root>/ACC.Enabled'
 * Inport: '<Root>/Target.Detected'
 */

rtb_targetdetected = ((AccelCtrl_ACC2_U.ACCEnabled != 0.0) &&
 (AccelCtrl_ACC2_U.TargetDetected != 0.0));

/* Switch: '<S4>/Switch1' incorporates:
 * Constant: '<S4>/Target dv'
 * Inport: '<Root>/Target.NearPoint_dv'
 */

if (rtb_targetdetected) {
    rtb_dvms = AccelCtrl_ACC2_U.TargetNearPoint_dv;
} else {
    rtb_dvms = AccelCtrl_ACC2_P.Targetdv_Value;
}
```
STUDYING THE SYSTEM BEHAVIOR FOR 40, 20, AND 0 KM/H.
HOW CAN WE MANAGE?

System Simulation:

Combine Plant and Controller Simulation
AGENDA.

1. Function Development in Automotive Industry
2. Integration of Functions
3. Improve Methods: Virtual System Prototyping
4. Role of Functional Mock-Up Interface
5. Example Formula Student Electric
ENTWICKLUNGSPROZESSE: IST-ZUSTAND.

- Co-Simulation Werkzeuge
- Propriätere Werkzeug-abhängige Schnittstellen

→ Lizenzen
MODULARISIERUNG UND STANDARDISIERUNG: FUNCTIONAL MOCK-UP INTERFACE.

- Standard für OEMs und Lieferanten

- Vereinfachter Werkzeug-unabhängiger Modelaustausch
  - Lizenzfrei
  - IP Schutz
OPENESS OF IT-SYSTEMS. CPO AND FUNCTIONAL MOCK-UP INTERFACE.

Presse-Information  
8. März 2012

Presseerklärung zur CPO-Pressekonferenz im Rahmen des AutomotiveDAY auf der CeBIT

Karl-Erich Probst, Leiter BMW Group IT: „Für die Entwicklung, die Produktion und den Vertrieb komplexer technischer Produkte wie unseren Premiumfahrzeugen ist eine große Anzahl spezialisierter IT-Systeme notwendig. Mit Hilfe des Product Lifecycle Managements (PLM) ist es in der Vergangenheit gelungen, diese heterogenen IT-Systeme und deren Daten zu vernetzen. Dabei ist eine der größten Herausforderung bei der Integration der IT-Systeme in die PLM-Landschaft, sicherzustellen, dass die "richtigen" Daten zur "richtigen" Zeit am "richtigen" Ort verfügbar sind.

Fehlende Offenheit der IT-Systeme stellt sich immer wieder als besonderer Hinderungsgrund für die effiziente Einbindung und damit als wesentlicher Kostentreiber für die IT heraus. Beispielsweise ist es für uns als Datenerzeuger

Link to Presseartikel AutomotivIT

Functional Mock-Up Interface is an implementation of the CPO.
FMI-based Initiative in Automotive Industry

- BMW, Daimler and Ford started in Q2 2012 an initiative to establish FMI as the standard for simulation model exchange between OEMs and suppliers.

- These OEMs as well as Chrysler, Fiat, GM, Jaguar Land Rover, Nissan, Renault, Toyota and Volkswagen signed on Oct 23rd/24th at GAAG conference a commitment to support this initiative.

- Accompanying “Smart Systems Engineering” project with OEMs, suppliers and IT tool vendors started in October at ProSTEP iViP.
FUNCTIONAL MOCK-UP INTERFACE.

- Tool-independent Exchange of Simulation models
- Currently Standard 1.0, developed by the MODELISAR ITEA2 project
- Further development as a Modelica Association Project: http://www.fmi-standard.org
- Current development: FMI – version 2.0 Beta 4
FUNCTIONAL MOCKUP UNIT.

• Functional Mockup Unit is a zip with extension .fmu, contains:
  • modelDescription.xml
    • Description of parameters, inputs, outputs
  • binaries/platform (e.g. win32)
    • Dynamic link library (optional static library)
  • and / or C-sources
  • Resources (data, images)
• Standardized access to model equations via FMI-routines
• FMI 1.0 – 2 options:
  • Model Exchange: without solver algorithm
  • Co-Simulation: with solver algorithm
• Two-step process:
  • Export of the FMU from the author tool
  • Import of the FMU into the target tool
MODEL EXCHANGE VS. COSIMULATION.

Enclosing Model
- \( t_0, p, \text{initial values (a subset of } \{x_0, x_0, y_0, v_0, m_0\}) \)
- \( t \) time
- \( m \) discrete states (constant between events)
- \( p \) parameters of type Real, Integer, Boolean, String
- \( u \) inputs of type Real, Integer, Boolean, String
- \( v \) all exposed variables
- \( x \) continuous states (continuous between events)
- \( y \) outputs of type Real, Integer, Boolean, String
- \( z \) event indicators

External Model (FMU instance)

Solver

Co-Simulation Master
- \( t_0, p, v_0 \)
- \( t \) time
- \( m \) discrete states (constant between events)
- \( p \) parameters of type Real, Integer, Boolean, String
- \( u \) inputs of type Real, Integer, Boolean, String
- \( v \) all exposed variables
- \( x \) continuous states (continuous between events)
- \( y \) outputs of type Real, Integer, Boolean, String
- \( z \) event indicators

Co-Simulation Slave (FMU Instance)
SOFTWARE TOOLS.

Integration Platform: CarMaker

- Dymola
- Simulation-X
- MATLAB / Simulink
- C / C++
FMU EXPORT FROM / IMPORT TO DYMOLA.

• **Export from Dymola:** built-in, no extra license
  
  `translateModelFMU(modelname, storeResult, modelIdentifier, fmiType, includeSource)`
  
  • ME and CS
  • CS only with Sundials CVODE (not available in Dymola)
  • Dymola license at target PC required (unless option Code Export is available)

• **Import to Dymola:** built-in, no extra license
  
  `importFMU(modelname, includeAllVariables, integrate)`
  
  • ME and CS
FMU EXPORT FROM / IMPORT TO SIMULATION-X.

• **Export from Simulation-X**: built-in, extra license
  • ME and CS

• **Import to Simulation-X**: built-in, no extra license
  • ME and CS
FMU EXPORT FROM / IMPORT TO C/C++.

• **Export from C/C++:**
  - Qtronic FMU Software Development Kit
  - ME and CS
  - BSD license

• **Import to C/C++:**
  - Qtronic FMU Software Development Kit
  - Modelon FMI Library
  - ME and CS
  - Both BSD license
FMU EXPORT FROM / IMPORT TO SIMULINK.

• **Export from Simulink:** Real-Time Workshop target rtwsfcnfmi
  • Provided by Dassault Systemes, no extra license
  • Requires Microsoft Visual C/C++ compiler
  • Only ME 1.0
  • Not all Simulink-blocks supported
  • Problems with parameters
  • → Modelon FMI Toolbox for MATLAB will be extended: FMIT Coder

• **Import to Simulink:** Modelon FMI Toolbox for MATLAB
  • ME and CS 1.0
  • Windows 32 and 64 bit, Linux 32 and 64 bit
  • License required
FMU IMPORT TO CARMaker.

- Modelon FMI Toolbox for CarMaker
  - ME and CS 1.0
  - Stand-alone tool, builds CarMaker application
  - License required
EXPERIENCE WITH…

… FMU running in real-time environments like HIL?
… FMU implemented by Hardware?
… simulating many FMUs because of stability?
… handling large number of inputs, parameters and outputs?
### AGENDA.

1. Function Development in Automotive Industry
2. Integration of Functions
3. Improve Methods: Virtual System Prototyping
4. Role of Functional Mock-Up Interface
5. Example Formula Student Electric
DEMONSTRATION OF POSSIBILITIES OF NEW TOOLS.
VIRTUAL SYSTEM PROTOTYPING. COMBINING NEW METHODS AND NEW TOOLS.

1. Virtual System Design: Modeling & Designspace Architecting
2. Virtual System Integration: Simulation & Designspace Evaluation
3. Validation & Decision

Methodology

Physical Models
- Authoring Tool 1: Dymola
- FMU: powertrain
- Integration Platform: IPG CarMaker

Signal-oriented Models
- Authoring Tool 2: MATLAB/Simulink
- FMU: controller algorithm
- Authoring Tool 3: N.N.
- FMU: powertrain
VIRTUAL SYSTEM DESIGN.
VIRTUAL SYSTEM INTEGRATION.

1. Definition of FMU with signal-oriented, CarMaker-specific interfaces

2. Converting physical interfaces into signal-oriented interfaces with modelica sensor component

3. Synchronizing axes with modelica speed component
VIRTUAL SYSTEM INTEGRATION.

4. Exporting FMU using the Modelon FMI Toolbox

5. Integrating the powertrain model using the OpenXWD framework

→ Use of „Integration Substeps“ with Euler solver possible
TEST OF NUMERICAL STABILITY.

- Analysing error with different step sizes
- Mean square error over step size shows euler convergence in stabilized area
DESIGN EVALUATION.

How does the variation of Engine-Power influences the 3 FSE competitions?

Acceleration:

Skidpad:

Endurance:
DESIGN EVALUATION.

Acceleration:

Skidpad:

Endurance:
## DESIGN EVALUATION.

<table>
<thead>
<tr>
<th></th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor-Speed [rpm]</td>
<td>7600</td>
<td>6000</td>
<td>4000</td>
<td>4320</td>
<td>3300</td>
<td>2900</td>
<td>2300</td>
</tr>
<tr>
<td>Motor-Torque [Nm]</td>
<td>90</td>
<td>80</td>
<td>80</td>
<td>64</td>
<td>60</td>
<td>43</td>
<td>32</td>
</tr>
<tr>
<td>Motor-Power [kW]</td>
<td>76.5</td>
<td>53.6</td>
<td>34.8</td>
<td>30.1</td>
<td>22.2</td>
<td>14</td>
<td>8.3</td>
</tr>
<tr>
<td>Acceleration-Time [s]</td>
<td>4.62</td>
<td>4.91</td>
<td>5.31</td>
<td>5.67</td>
<td>6.2</td>
<td>7.32</td>
<td>8.91</td>
</tr>
<tr>
<td>Skidpad-Time [s]</td>
<td>5.29</td>
<td>5.29</td>
<td>5.29</td>
<td>5.29</td>
<td>5.29</td>
<td>5.29</td>
<td>5.29</td>
</tr>
<tr>
<td>Endurance-Time [s]</td>
<td>60.3</td>
<td>60.3</td>
<td>60.5</td>
<td>60.5</td>
<td>60.9</td>
<td>62.12</td>
<td>66.2</td>
</tr>
<tr>
<td>Endurance-Consumption [kWh]^2</td>
<td>2.93</td>
<td>2.93</td>
<td>2.76</td>
<td>2.82</td>
<td>2.53</td>
<td>2.24</td>
<td>1.76</td>
</tr>
</tbody>
</table>

- **M5** with reduced weight by 20 kg at 274 kg total weight

### Graphs:

- **Skidpad**
- **Acceleration**
- **Endurance**
- **Endurance**
PROSPECT.

- Systematic Evaluation with Design of Experiments
- Localize optimized Trade-Offs
THANK YOU.
AGENDA.

1. Function Development in Automotive Industry
2. Integration of Functions
3. Improve Methods: Virtual System Prototyping
4. Role of Functional Mock-Up Interface
5. Example Formula Student Electric
ANY QUESTIONS?