Towards systematic Model and Component based development of embedded systems

Martin Törngren, martin@md.kth.se
Embedded Control Systems group,
Division of Mechatronics – Department of Machine Design
KTH - Royal Institute of Technology, Stockholm, Sweden

www.md.kth.se/RTC
Embedded Control System group

Systems engineering for embedded (control) systems

Research themes:
- Architectural design
- Model based development
- Systematic verification and integration

KTH Embedded Systems Centre
Projects: ARTIST2, ATESSST, DYSCAS, Rembrandt, +
Current emphasis on automotive applications

See www.md.kth.se/RTC for more info
From Mechanics to Mechatronics
Mechanical engineering CBD and MBD

• Well developed methodologies
  However, functional abstractions are not used in practice!
• Component attributes and interfaces
  - Design, look and feel – also interpreted as cognitive interfaces (cognitive) in research
  - Geometry,
  - Physical effects and dependencies – a challenge
• Mature adoption of MBD
  - Technical engineering – CAE
  - Information management – PDM
• Trend
  - Extend systems analysis, introduce functional abstractions / rationale, towards PLM
Model Data Management Platform

Existing or new data management tools

Modeling & analysis tools

Loose integration

Tool-independent format

Integration mechanism

Version Control

Change Management

Management functionalities

Data repository

Martin Törngren, KTH, 2007-09-30
Mechatronics: Adding new functions

Efforts: Software component and middleware standardization

Add complete control unit: SW, HW, often + sensors and some mechanics

Towards model level integration

Martin Törngren, KTH, 2007-09-30
**“Purely” mechanical vehicle**

<table>
<thead>
<tr>
<th></th>
<th>Sus/C</th>
<th>Brake</th>
<th>Steer</th>
<th>Wheel</th>
<th>Diff</th>
<th>Trans</th>
<th>Clutch</th>
<th>Eng</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Brake</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Steer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Wheel</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diff</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trans</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Clutch</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Eng</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Driver</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

* X - Mechanical relations
Fully programmable vehicle!

<table>
<thead>
<tr>
<th></th>
<th>Sus/C</th>
<th>Brake</th>
<th>Steer</th>
<th>Wheel</th>
<th>Diff</th>
<th>Trans</th>
<th>Clutch</th>
<th>Eng</th>
<th>Driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Susp</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td></td>
</tr>
<tr>
<td>Brake</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td></td>
</tr>
<tr>
<td>Steer</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td></td>
</tr>
<tr>
<td>Wheel</td>
<td>X</td>
<td>X</td>
<td>X+P</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diff</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>X+P</td>
<td>P</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trans</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>X+P</td>
<td>P</td>
<td>P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clutch</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>X+P</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eng</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Driver</td>
<td>P</td>
<td>X+P</td>
<td>X+P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>P</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

P - Programmable relations
X - Possible change
Additional complexities

New functions and technologies
- Domain convergences

Product lines, platforms and components shared across products
- Varying component contexts; "No” optimal architecture

Conflicting requirements, many stake-holders
- Shared design parameters and trade-offs
- Application level vs. Platform level solutions
  - Optimality, compare end to end argument
  - Combined synthesis?

Emerging behavior
- Implementation dependencies
- Systems integration (uncontrolled behavior)
- External behavior
Dependencies illustrated for control systems

Source: Törngren et al. 2007, CACSD (see references)
Characterizing control/embedded systems co-design

[Diagram showing the relationship between digital controller, plant, computation, communication, timing, accuracy, memory, power, and other factors.]

Domain specific design parameters and qualities

Abstraction, constraints & views

Mapping

Analysis & synthesis

Digital controller

Plant

Platform

SW

HW

Computation

Communic.

Timing

Accuracy

Memory, Power, ...

Abstraction, constraints & views

Domain specific design parameters and qualities

Martin Törngren, KTH, 2007-09-30
Embedded systems scope

- Vehicle Level
- Analysis Level
- Design Level
- Implementation Level
- Operational Level

See Cuenot et al for more info and ATESS: www.atesst.org
Aspects

Levels

Functionality Performance Safety/reliability Reuse Maintenance …

Views

Components, level i

Platforms, level i

Components and Platform, level j

Refinement

Product

Models for communication, documentation, analysis and synthesis purposes. See Törngren et al 2005 for details.
Research directions?

Dedicated integration between tools
- E.g. Functions – performance, Functions – platform,
  Design models – safety, HW architecture – geometry, ...

Repository/information model based solution with integration to domain tools

Integration languages
- SW/HW scope: E.g. AADL and Autosar
- Trend: Broader embedded systems scope: EAST-ADL

Behavioral languages
- Trend: Capture more non-functional aspects, e.g. Metropolis

Evolving situation!
Many research and standardization efforts!
Model & component based development

Component technologies and composition theories

Technologies for model-based development
- Meta-modeling and qualities/aspects frameworks, dependencies, transformations, analysis and synthesis, behavior and structure

Componentization context:
- Strongly related to architectural design and modular products
- Product lines – a structured approach to componentization
- System scenarios providing requirements for component models

Model-based development framework:
- Goals and drivers
- Context: Technology, processes, organizations, business
- Product abstractions, qualities, aspects and parts
- Communication, documentation, analysis, synthesis
- Means: Abstraction, Formalization& structuring, Visualization, Refinement, Prediction, Automation, Methodology
- Keys for Industrial adoption
Many nice (research) challenges!

- Component models for different abstraction levels;
  - Required and provided services
  - Required and provided qualities
  - Fault behavior
  - Functionality, reliability, timing/resource util., maintenance, flexibility
- Application / middleware / platform trade-offs
  - Methodology
  - Synthesis and platform based design
- Reconciliation of model languages
  - Ontologies and View models
- UML – how far can it be formalized and adapted for embedded systems engineering?
  - UML behavior models
  - Safety analysis
  - Continuous-time systems
Complementary information

A complementary presentation, including case studies, was given at the DATE – ARTIST2 workshop
http://www.artist-embedded.org/artist/-ARTIST2-Workshop-at-Date-07-.html

Research projects underlying this presentation include

- ATESST: www.atesst.org
- SAVE, Modcomp, ARTIST2 – see links available here:
  - www.md.kth.se/RTC
  - See also the ARTIST2 Control for Embedded Systems Cluster through the ARTIST2 web-site.

Acknowledgements

- The work has been supported in the context of the above projects, funded by Vinnova, SSF the European commission and Scania.

Recent underlying references – see enclosed slides
References continued - papers


References continued – journals and book chapters


Web links to the corresponding books:
- http://www.springer.com/west/home/computer?SGWID=4-146-22-173752106-0
References continued – related theses


