Multiparadigm modeling – A Mechatronics and Embedded Control Systems perspective

Martin Törngren
Embedded Control Systems group
Mechatronics, Department of Machine Design
School of Industrial Engineering and Management
KTH - Royal Institute of Technology, Stockholm, Sweden

Outline

- Mechatronics
  - Opportunities and complexity
  - Human in the loop – who decides?
  - Embedded control systems
  - Codesign and Architecture design
  - Automotive embedded systems as an example
  - Dealing with complexity, approaches and trends
  - Industrial challenges
- Model based development in Mechatronics
  - Needs and challenges
- Research topics and challenges
Embedded Control System group

www.md.kth.se/RTC

Embedded control systems science and engineering

Research themes:
- Control and computer system co-design
- Architectural design, trade-off analysis
- Model and component based development
- Methodology
  - Cost-efficient systematic design and verification

ArtistDesign, Artemis, ICES

Current topics

- Self-configuring automotive embedded systems
  - Variability, Config. management, meta-data, QoS
- State of the art and evaluations
  - Modelica, Simevents, Simulink, Sysml/UML incl. Parametrics, Ptolemy, Bond graphs, VHDL-AMS,
  - Modeling languages for embedded systems
  - Architecture trade-off analysis techniques
  - Model transf. Techniques, survey
  - Methodologies
- EAST-ADL – Architecture description language for embedded systems
- Tooling (Eclipse plug-ins)
Integrating multiple views

Concept
- Multi-view modeling
- Model & tool integration
- Model data management
- PDM & SCM functionality

Instances
- PDM based
  - Simulink, Dome, configuration management
  - Build and SW/HW configuration management
- UML-based

The context: Mechatronic products

- Domestic robot
- Medical
- Embedded Control Systems
- Combustion engine
Mechanics → Mechatronics

Hy-Wire från GM
Skateboard concept
(Autonomy 2)

Fuel-cell
Distributed control
Electrical actuators

Existing electronic braking system

➢ Distributed components for control and actuation.
  - Several interfaces between pneumatics and electronics

Source: Haldex
**Future System (X-by wire)**

- Wheel end modules with integrated control system
  - Less interfaces and system components

**Mechanics vs. Mechatronics; adding flexible information processing and flow**

Source: Haldex

CAMPaM 2009, Martin Tömgren
### "Purely" mechanical vehicle

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

X - Mechanical relation

---

### Fully programmable vehicle!

<table>
<thead>
<tr>
<th>Susp</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>
</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>
</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>
</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>
</tr>
<tr>
<td>Diff</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>X+P</td>
<td>P</td>
<td>P</td>
<td>P</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>
</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>P</td>
<td></td>
</tr>
<tr>
<td>Eng</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>P</td>
<td>X+P</td>
<td>P</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>
</tr>
</tbody>
</table>

P - Programable relation
X - Possible change
Complexity in everyday devices
Embedded systems permeate life cycle
Issues: Cost-efficient dependability, flexibility and Maintenance

Mercedes removed 600 EE functions (2003)
Costs for development of electronics/SW is today reaching about 40% of total costs [Gri03].
Development of complex products

Model Based Integration

- Organization
  - Competence
  - Responsibilities
  - Communication

- Process
  - Analysis
  - Modelling/tool use
  - Synthesis/decisions

Social Integration

Product

- Function
- Implementation

Architecture Integration

Automotive embedded systems and model based engineering

Specifics:

- Tightly coupled systems: Software, electronics, mechanics
- Hybrid behaviors, timing and criticality “levels”
- Emergent behavior more difficult to control and understand due to
  - Information and physical interactions
  - Systems integration (uncontrolled behavior)
  - Connection with external behavior

- Multiple tools, modeling languages and standards
- Well established disciplinary/subsystem MBE
  - Control, Electronics, Mechanics
- Weak MBE for systems architecting and integration
- Weak MBE for software engineering; Autosar is pushing a change
Automotive embedded functionality

- Automatic toll
- Air Con
- Light control
- Alarm detection
- Panel
- Sleep detector
- Engine control
- GPS
- Airbags
- Gearbox
- Radar
- Clutch
- ABS
- Supension

Global Coordination: Gerard Berry, Artist summer school

Types of behaviors of automotive embedded systems

Global Coordination: Calc+CC+FSM

Source: Gerard Berry, Artist summer school
Example integration (Far project)

Characterizing control/embedded systems co-design
Example co-design tool-set: AIDA

1. Control design in Simulink
2. Import the control design to the AIDA tool-set
   Import
3. Define mapping
   Define mapping
   Export
4. The resulting control design with embedded analysis results is exported to Simulink. The control performance can be analysed through simulation.

Architecture analysis (e2e timing)

Co-design and relations illustrated for control systems

Legend:
- Components: Design choices
- HW/Platform incl. processors and network
- SW platform including com. protocol
- Control design, algorithms & structure
- Basic closed loop dynamics
- Basic execution and com. time
- Task level behavior (utilization, response times)
- Control performance (variance, rise time, overshoot,..)
- Mapping decisions: Functions to processors, signals to com. means, and types
- Quantization
- Determines

Tomgren et al. 2007, CACSD (see references)
Research at the Mechatronics lab/KTH

Specifications

System design (behavior, structure)

Integration
Components, tools, people and processes

Information management
- PDM/STEP-based
- OMG/MOF-based
- SCM tools
- Embedded systems specific ("non" standard)

Technical process
- Requirements engineering
- Behavior modeling/sim.
- Architecture design
- Safety engineering
- CAD, cabling harness
- Thermal, power, ...

• Open frameworks & environments (e.g. Eclipse, GME)
• Technologies (metamodeling, formats, transformations, analysis, tools) and Standards
• Evolving area!
Automotive Embedded systems and MBE

- Limited usage of UML/SysML
  - Subsets, e.g. sequence diagrams (specs, testing), structure diagrams (architecture), software design
  - Meta-/information-modeling, for example in Autosar

- Many efforts (research and industrial) in furthering UML/SysML for Embedded systems modeling
  - Domain specific modeling languages; EAST-ADL, AADL
  - Projects, e.g. ATESST2, CESAR, Modelisar, Speeds, SysModel and Timmo
  - Safety related standards are studied to incorporate considerations for MBE (e.g. ISO26262)

EAST-ADL in a nut-shell

A non traditional ADL in the sense that it covers much more:
- Variability
- Requirements
- Safety
- Behavior
- Environment Modelling
- Design methodology

Concrete results
- Domain model and UML2 Profile
- Prototype Tool
- Active Safety Demonstrator
- Examples and case studies

EAST-ADL has been developed in: EAST-EAA, ATESST and ATESST2.
Now being consider in CESAR.

Coordinated by Volvo Automotive, Suppliers Tool vendors Universities including KTH

Alignment/integration:
- SysML, AADL, AUTOSAR
- ISO26262
- IST/EUCAR Active Safety

www.atesst.org
Objectives

- EAST-ADL provides means to represent the embedded system in several abstraction levels
- Different kinds of engineering information:
  - Requirements
  - Feature content
  - Functional content
  - Variability
  - Safety information
  - V&V Information
  - SW architecture (AUTOSAR)
- Traceability
- Analysis and synthesis
- Document generation and views

Relation to other modeling languages and approaches?

Why Not UML?
- EAST-ADL is implemented as a UML2 profile, adding properties for embedded systems.

Why not SysML?
- EAST-ADL is a specialization of applicable SysML concepts

Why not Autosar?
- EAST-ADL complements Autosar with for example functional structure and safety properties

Why not proven proprietary tools (Simulink, Statemate, Modelica, ASCET, ...)?
- ATESTT integrates external tools and provides an information structure for the engineering data regardless of tool
EAST-ADL and domain specific tools - focusing on behavioral modeling

EAST-ADL and Simulink

**EAST-ADL**
- Information structuring
- Predefined views and levels of abstraction
- Constitutes an integration point
- Focused on structure but provides also native behavior

**Simulink**
- Behavior modeling and simulation
- Modeling at different levels of abstraction
- Behavior focused; many models of computation
- Logical structuring for
  - model management
  - code generation
  - SW components

Many technical issues! Semantics in the transformation!
EAST-ADL support for safety engineering

- **Error Modeling** - capturing and managing potential errors and propagations at different levels of abstraction. I-FMEA
- **Safety Analysis** - inferring causes and consequences of errors through external tools. FTA
- **Safety Requirements** - specifying hazards and subsequent function and quality requirements.
- **Safety Case Method** - arguing that a system is acceptably safe in a particular context.

Challenges

- Still a long way to go for full model based (embedded) systems engineering
- Information and model management
  - Lack of CM on model level
  - Lack of unified way of handling software (SCM) and hardware (PDM)
- Lack of workable exchange formats
  - XMI/DI, AP233
  - Specific ones exist
    - Network messages (CAN, Flexray, ...)
    - Debugging and calibration formats (ASAM/ASAP)
- Methodologies and training/guidelines
- CBD + MBD: Components with meta-data; contracts defining assumptions for reuse
  - Compositionality!
Challenges

• Requirements -> Test cases
• Incorporation of “formal” analysis and verification for embedded systems
  Connecting design and analysis models
  - Simulink and Safety
  - Architecture and timing analysis
  - Nominal and error behavior
• Code generation as a synthesis problem
  - Automated decision making; Impact on actual behavior
• DSL solutions
  - Require multiple domain experts
  - Meta-modeling and transformation experts
  - Rapid prototyping environments (evolving DSL)

Suggestions for CAMPaM topics

• Formalism taxonomy / framework
  - Multitude of existing and new “formalisms” and supporting technologies!
  - Terminology
    - Behavior, structure, properties
    - Design context: LoA, qualities, analysis, tasks
    - Technology: E.g. different types of model transformations
• Model/tool integration and management
  - Mechanics, electronics, software, systems
  - Roadmap
  - Bottlenecks, research focus
  - Industrial and scientific perspectives