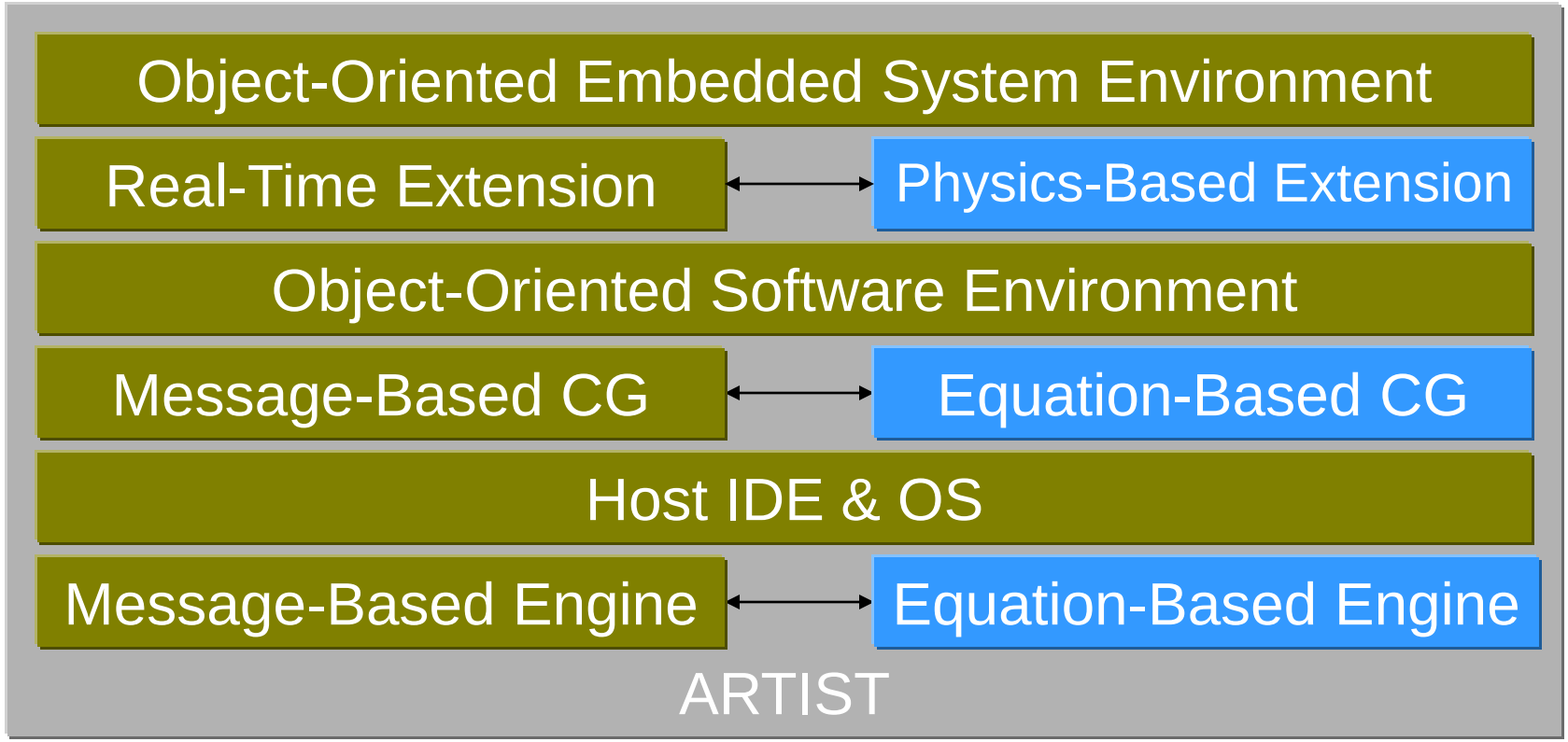
ARTIST Anatomy

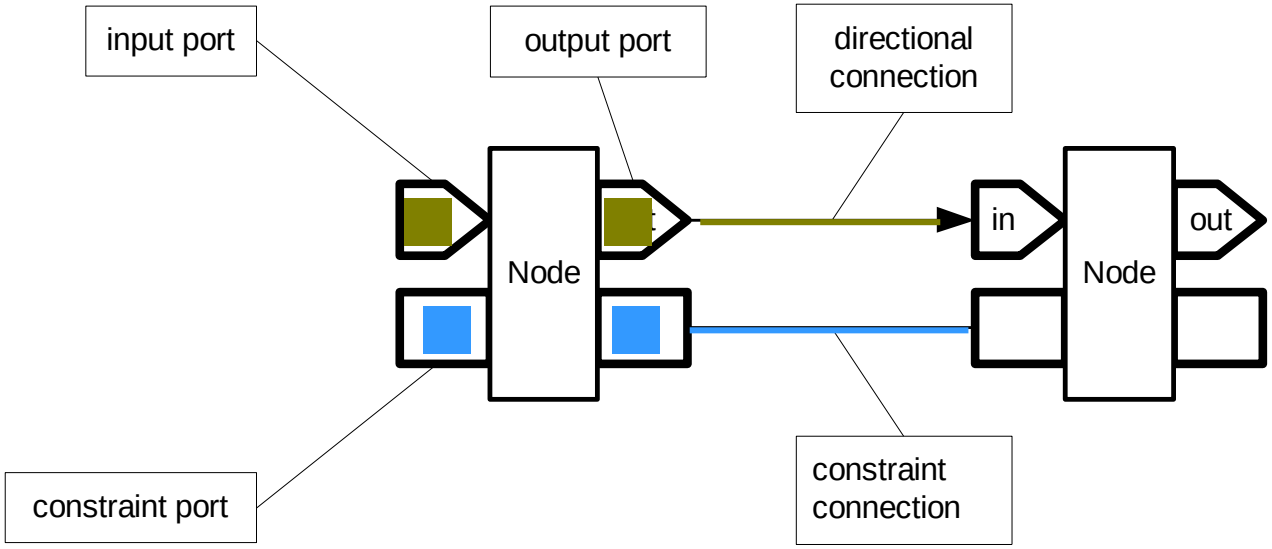


Partner Product (Rational, Telelogic, I-Logix, Artisan Software, Microsoft)

Open Numerics Product

Embedded System *Development* Environment

- Model-Based system analysis, design, integration and testing – same objects
- Requirement management, traceability, impact-analysis – same objects

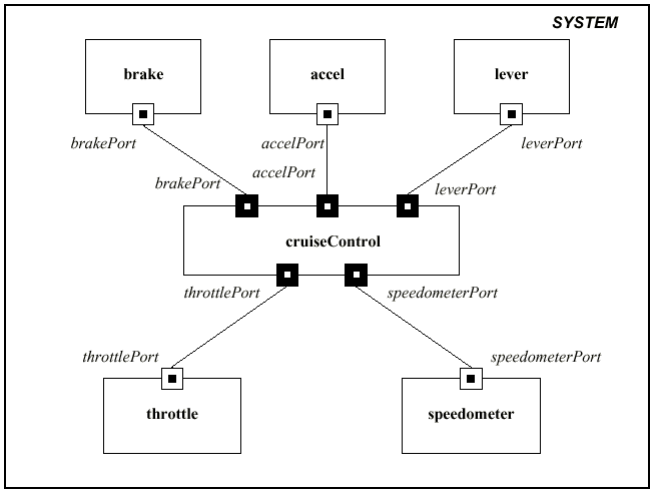


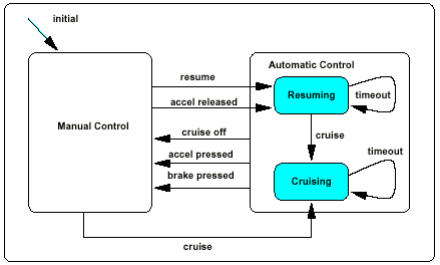
Real-Time Sequential Process (Data and Events) Communications

Real-Time Physics-Based (Constraints) Communications

Real-Time Extension

- Real-Time Models of Computation (Ptolemy – Ed Lee)
- Real-Time Object-Oriented Modeling (ROOM)
- UML-RT and UML 2.0
- Support A&D, Testing, RTOS



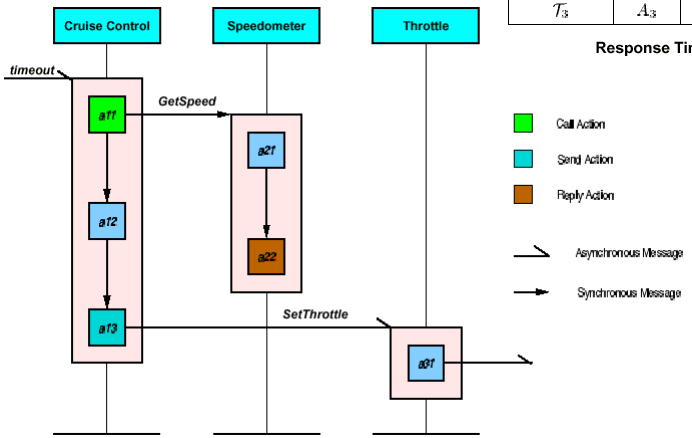


Transaction	Action	Response Time	Action	Response Time
$\mathcal{T}_1$	A_5	295	A_4	719
$\mathcal{T}_2$	A_2	382	A_8	590
$\mathcal{T}_3$	A_3	580	A_{11}	416

Response Times for Single Threaded Implementation

Transaction	Action	Response Time	Action	Response Time
$\mathcal{T}_1$	A_5	27	A_4	156
$\mathcal{T}_2$	A_2	64	A_8	156
$\mathcal{T}_3$	A_3	141	A_{11}	570

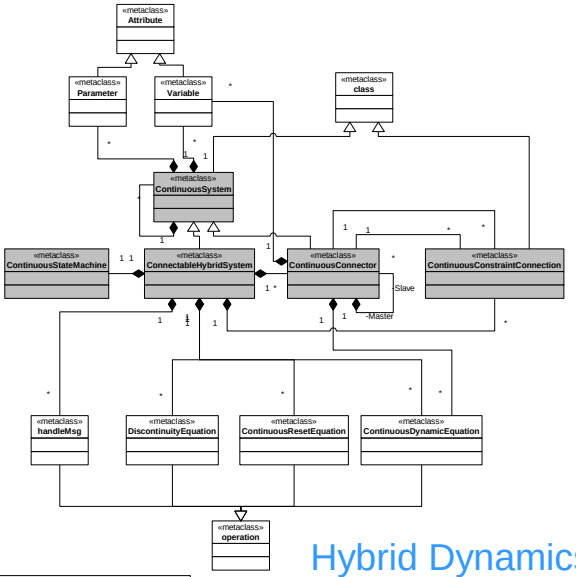
Response Times for Multi-Threaded Implementation



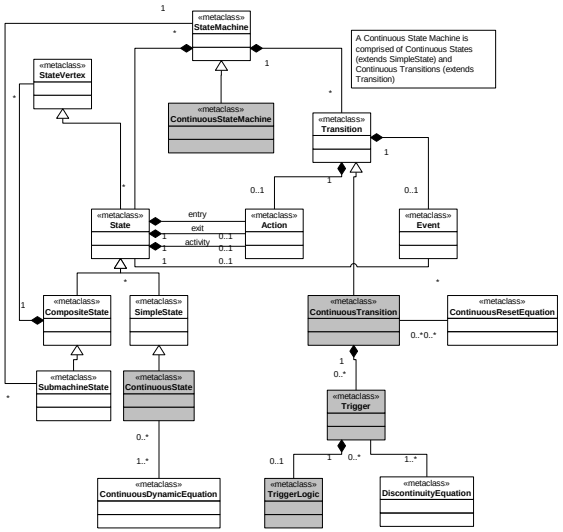
"Designing for Schedulability Integrating Schedulability Analysis with Object-Oriented Design", M. Saksena and P. Karvelas.

Physics-Based Extension: Overview

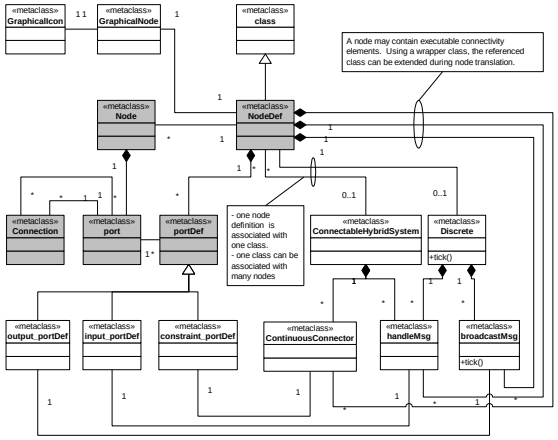
- Real-time software analysis and design
- Concurrent engineering
- Reusable and plug-n-play
- Inspired by Modelica and Simulink
- Based on UML-RT Extensibility



Hybrid Dynamics Model

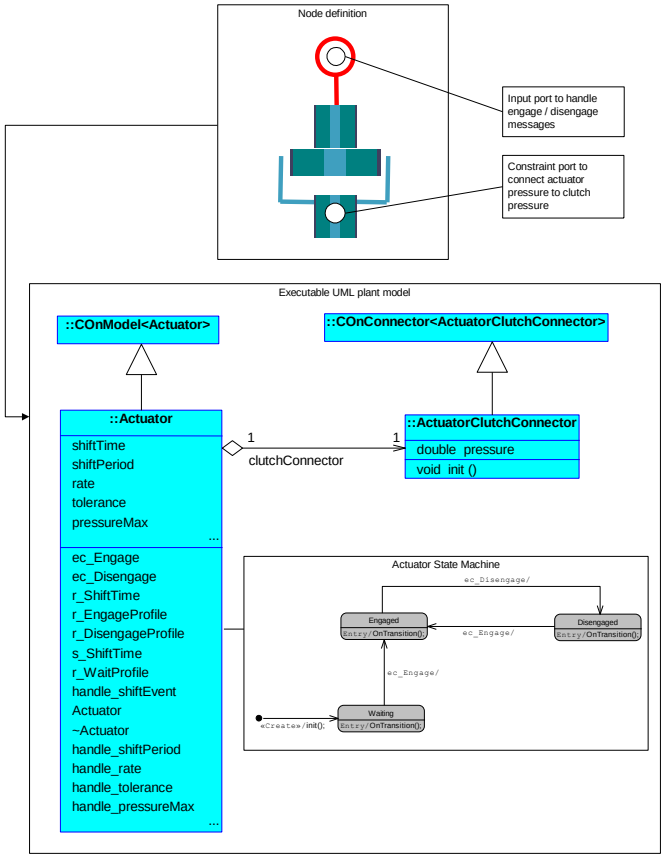


Continuous State Machine Model

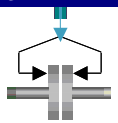
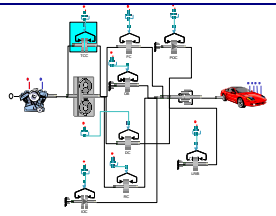
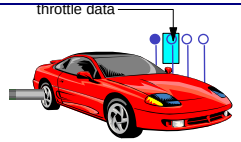
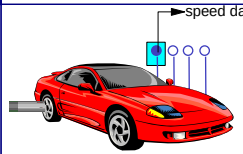


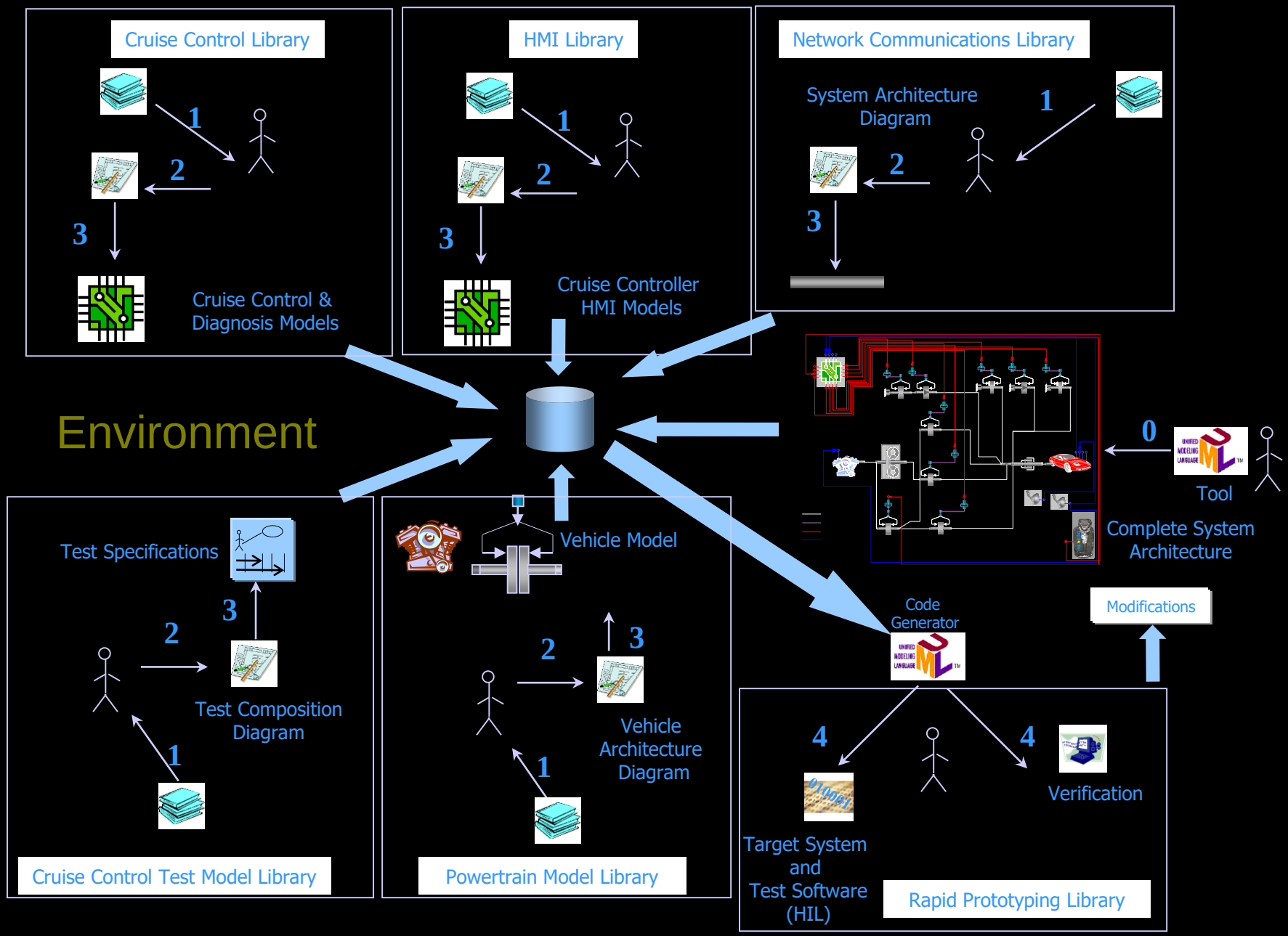
Hybrid Node (Capsule) Model

Physics-Based Extension: Node Definition



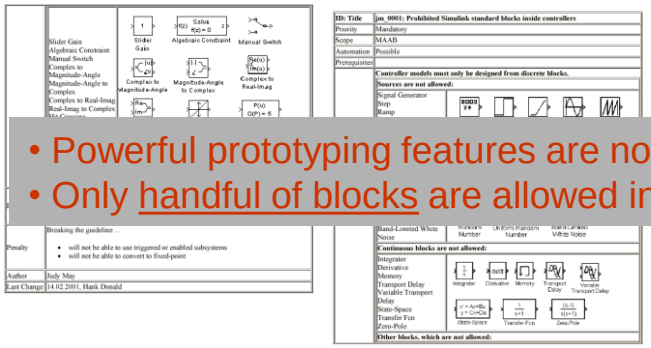
Clutch Actuator

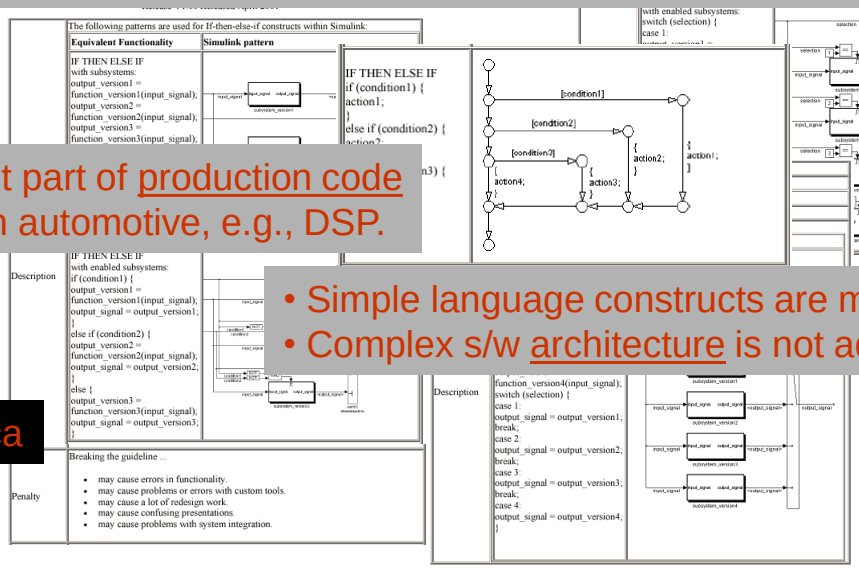
View element	Model element	Example
Node definition	UML Class	
Node (node definition instance)	Object (class instance)	
Input port	Message handler operation	
Output port	Message broadcast object	
Constraint port	Continuous connector	


Cruise Control Library
1
2
3
Cruise Control & Diagnosis Models
HMI Library
1
2
3
Cruise Controller HMI Models
Network Communications Library
System Architecture Diagram
1
2
3
Environment
0
Tool
Complete System Architecture
Test Specifications
3
2
Test Composition Diagram
1
Cruise Control Test Model Library
Vehicle Model
2
3
Vehicle Architecture Diagram
1
Powertrain Model Library
Code Generator
Modifications
4
4
Verification
Target System and Test Software (HIL)
Rapid Prototyping Library

Message-Based CG: Prototyping vs. Production Gap

Option: Move development to a prototyping environment.





- Powerful prototyping features are not part of production code
- Only handful of blocks are allowed in automotive, e.g., DSP.

- Simple language constructs are made complex
- Complex s/w architecture is not addressed

CONTROLLER STYLE GUIDELINES FOR PRODUCTION INTENT USING MATLAB ®, Simulink ® and Stateflow ®

North America

2.1. Motivation

The MAAB guidelines are an important basis for project success and teamwork - not only in-house, but also when cooperating with our partners or subcontractors.
Respecting the guidelines is one key prerequisite to achieving ...

- system integration without problems.
- clean interfaces.
- uniform appearance of models, code and documentation.
- reusable models.
- readable models.
- hassle-free exchange of models
- avoidance of legacies.
- a clean process
- professional documentation.
- understandable presentations.
- fast software changes.
- cooperation with subcontractors.
- handing over of (research or predevelopment) projects (to product development)
- ...

this means functionality, quality, safety, reduced time-to-market and cost reduction !
Disclaimer:
The MAAB guidelines are still under construction, resulting in frequent additions and changes of guidelines.
Please check regularly for updates of the guideline document!

Europe

State of the Practice: Methods and Tools (1)

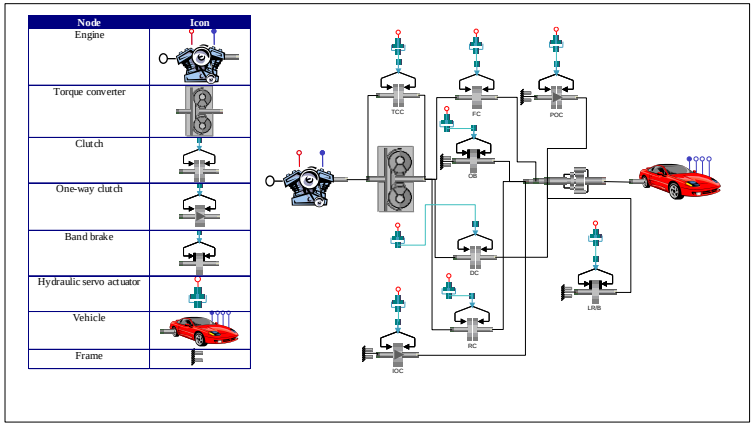
Development Phases:

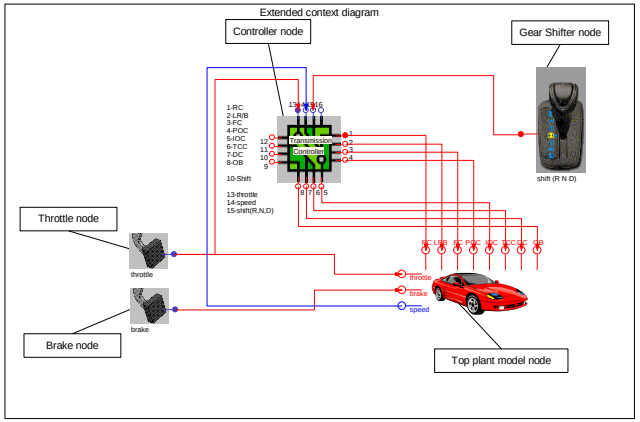
- System and SW requirements capture
  - still widespread: MS-Office
  - swing to explicit RE-tool (e.g. Telelogic Doors)
- System and SW design
  - for control theory applications: Matlab / Simulink / Stateflow
  - quite widespread in general: ASCET-SD
  - in future: UML-tools (e.g. Rose RT, Rhapsody), but not only for typical OO-applications like infotainment

State of the Practice: Methods and Tools (2)

- SW implementation
  - code generation with fairly optimized code (e.g. ASCET-SD)
  - for infotainment and telematics applications: UML-tools (e.g. Rose RT and Rhapsody)

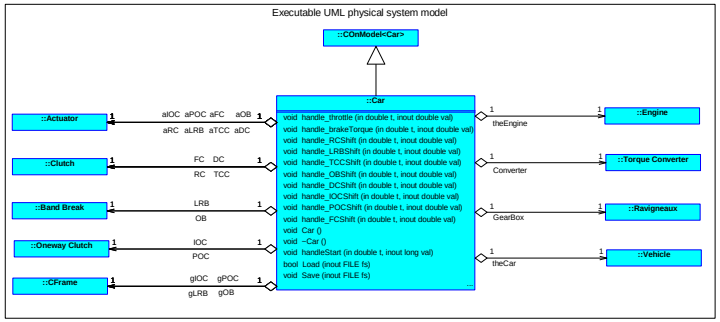
Equation-Based CG: Extended Context Diagrams

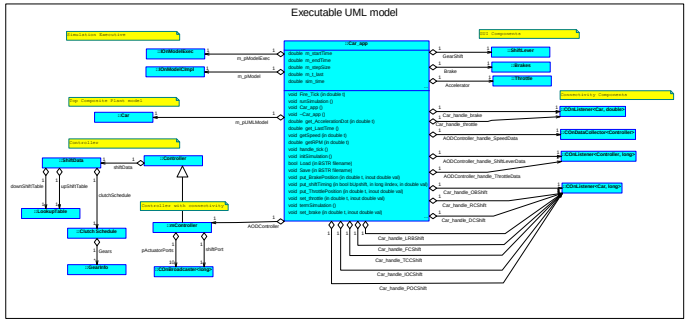




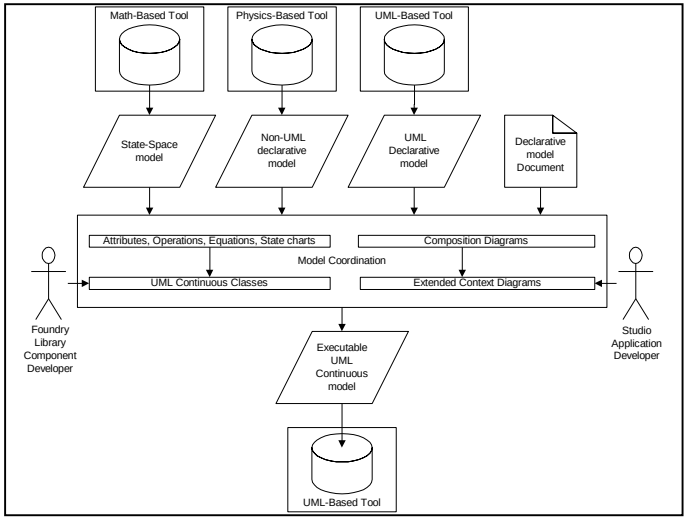
Translate to executable UML physical system model

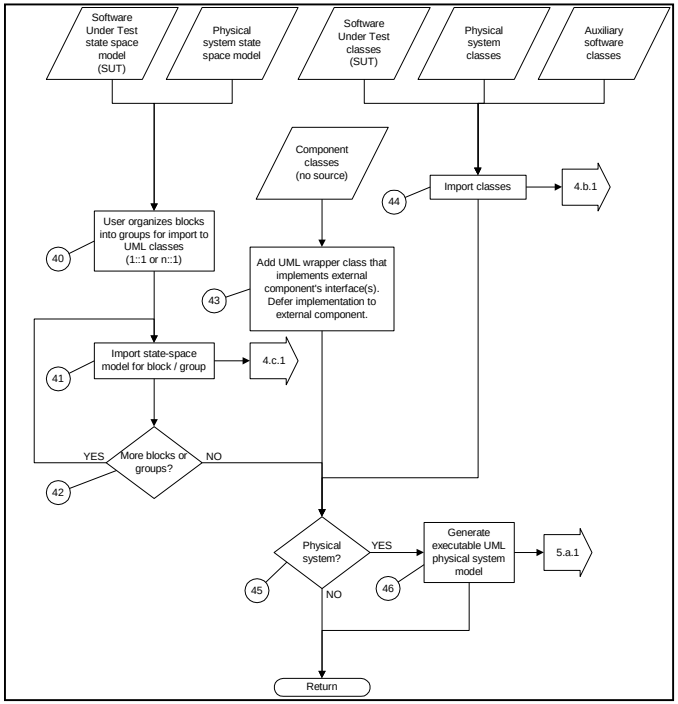
Translate composition to executable UML application model





Equation-Based CG: Import Models





Equation-Based Engine

- Solvers (ODE, High-Index DAE, Explicit, Implicit, NLA)
- Graph algorithms (Assignment, BLT, Index, Initialization)
- Linear Algebra (LAPACK and direct sparse)
- Event Detection (Interval ZC, Chatter-Control, Re-Initialization)
- Auto-Differentiation
- Event management and access interface
- Snapshot control
- Automation Interface
- Waveform Relaxation
- RTOS schedulable