

SysML for Systems Modeling

Matthew Hause – Artisan Software

Franco Fummi



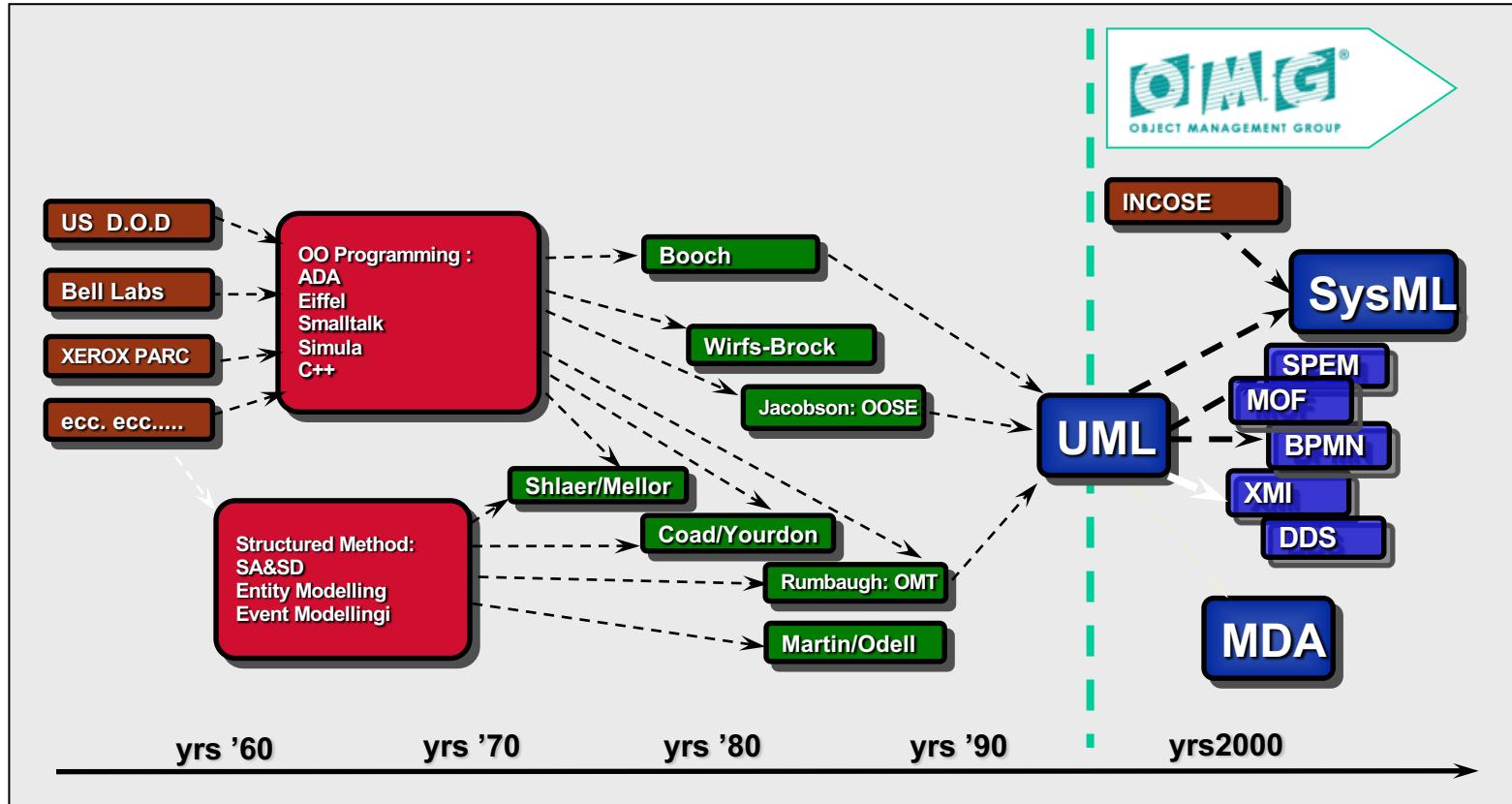
UNIVERSITÀ
di VERONA
Dipartimento
di **INFORMATICA**

Version 1.0

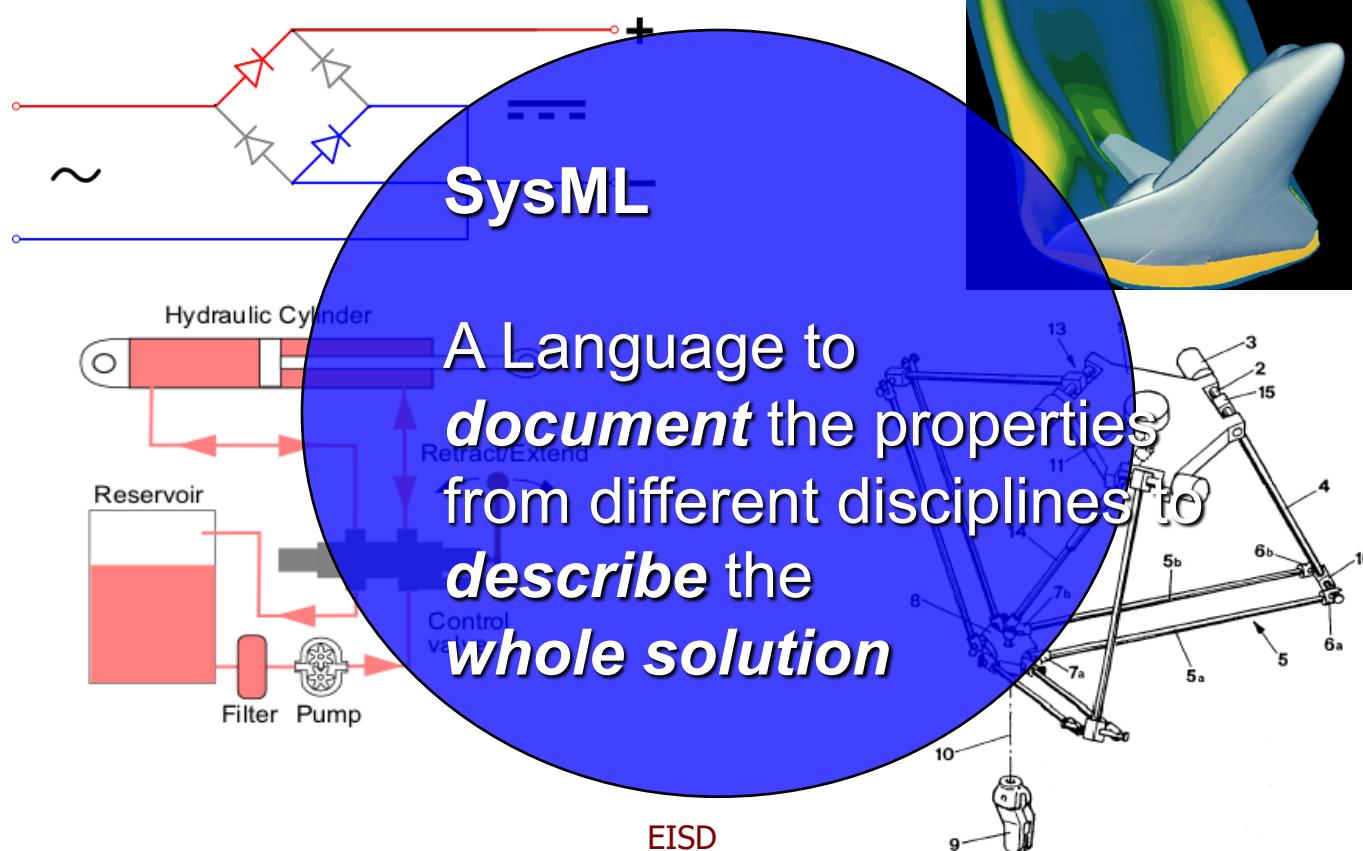
Topics

- SysML Status
- SysML Language Architecture
- SysML Language Detail
 - Requirements
 - Structure
 - Parametric Models
 - Allocation
- Summary

A Historical Perspective



A Unifying Systems Language



SysML Contents Summary

Varie viste del progetto:

- **Structure**
 - e.g., system hierarchy, interconnection

Fondamentale per vedere la struttura e funzionamento di ogni componente (in base alle svolte)
- **Behaviour**
 - e.g., function-based behaviour, state-based behaviour

specifiche degli stati che possono descrivere come un sistema può reagire alle varie condizioni
- **Properties**
 - e.g., parametric models, time property

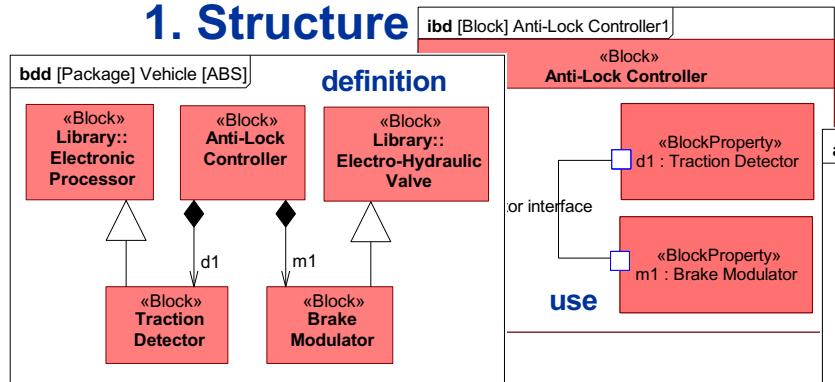
→ Significativa per passare da SysML a MARTE per rappresentare sistemi real-time
- **Requirements**
 - e.g., requirements hierarchy, traceability
- **Verification**
 - e.g., test cases, verification results

Non lasciare spazio per implementazioni non verificate che il progetto sia giusto
- **Cross-Cutting**
 - e.g., allocation of entities for workflow

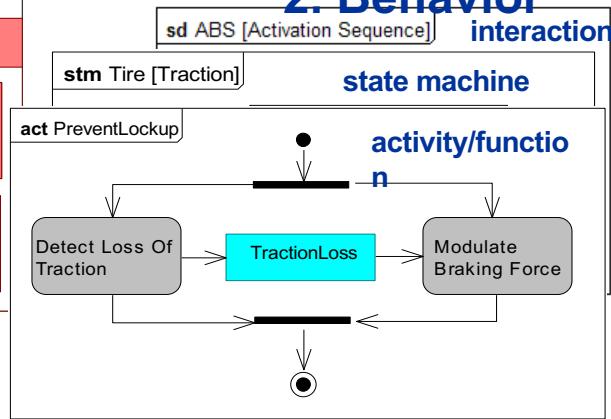
Esempio: test, ma dovranno essere stati fatti i test su ogni sotto-entità e non solo su entità complessa

The Four Pillars of SysML (ABS Example)

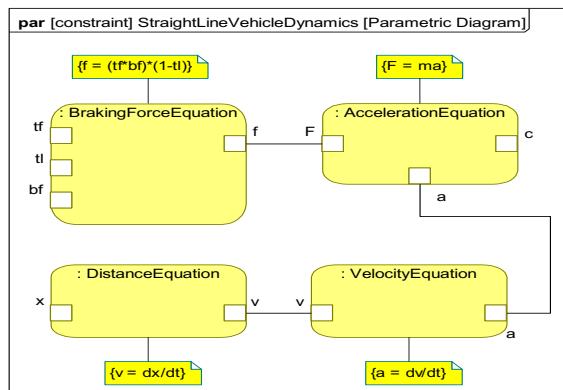
1. Structure



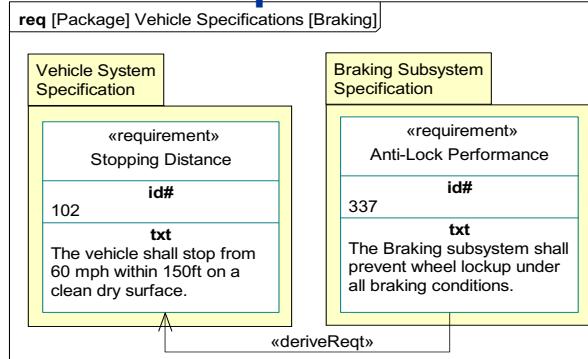
2. Behavior



4. Parametrics



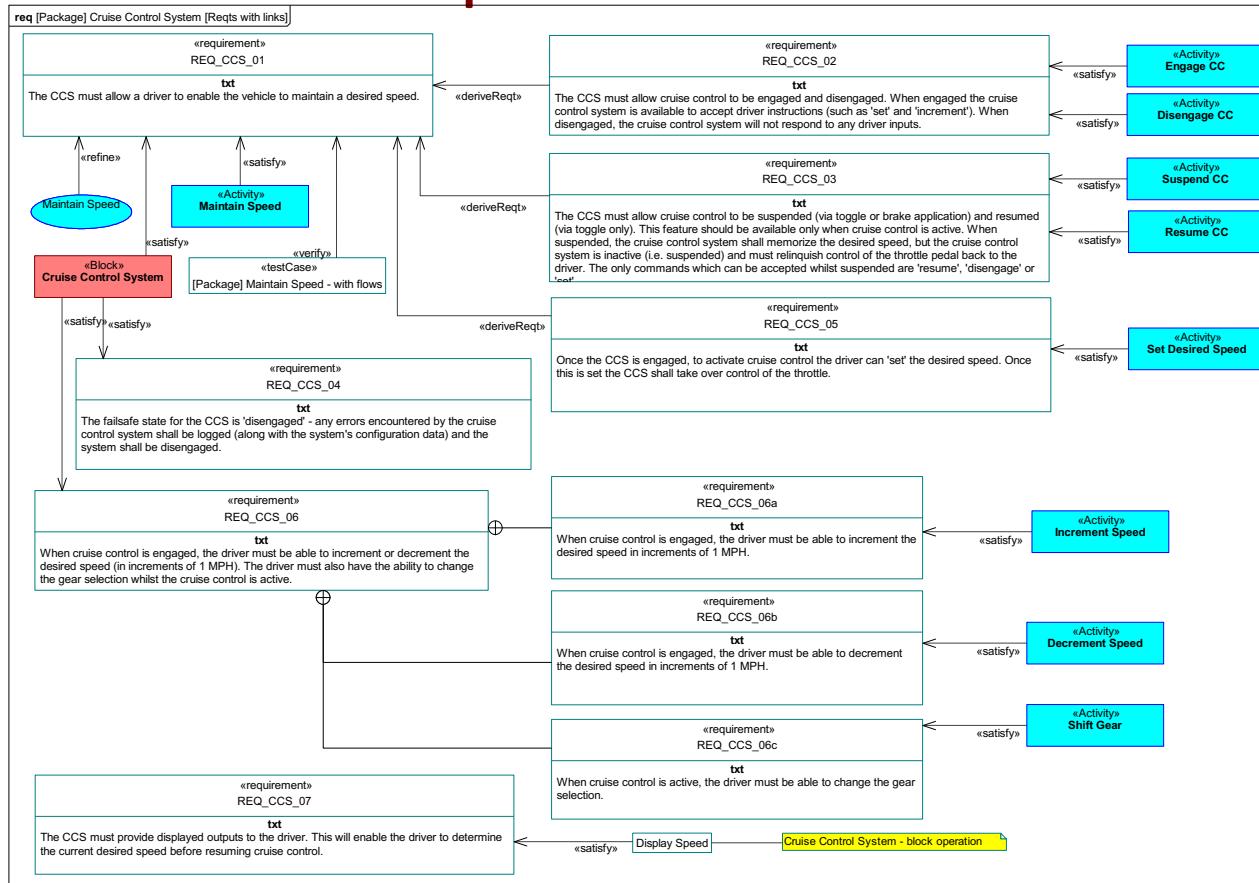
3. Requirements



Requirements

- Requirements represents a text based requirement
 - Includes id and text properties
 - Can add user defined properties such as verification method
 - Can add user defined requirements categories (e.g. functional, interface, performance, ...)
- Requirements hierarchy describes requirements contained in a specification
- Requirements relationships include DeriveReqt, Satisfy, Verify, Refine, Trace, Copy
- Graphical, tabular and tree notation specified

User Requirements Flow Down

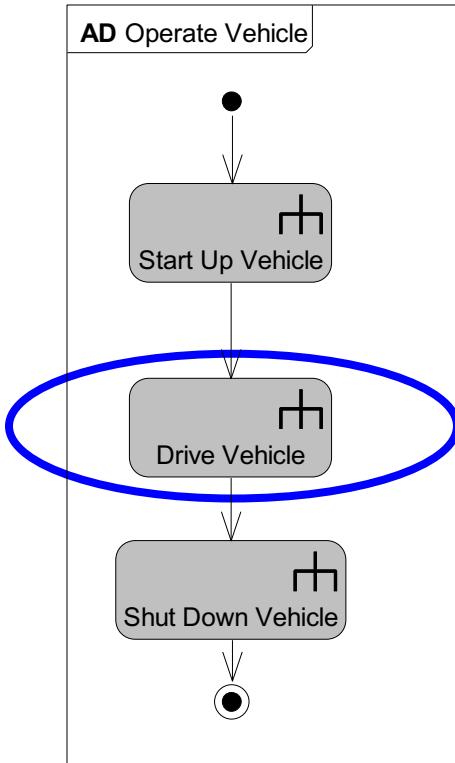


Activities

- Activity used to specify the flow of inputs/outputs and control, including sequence and conditions for coordinating activities
- Secondary constructs show responsibilities for the activities using swim lanes
- SysML extensions to Activities
 - Support for continuous flow modeling
 - Support probabilistic choice
 - Alignment of activities with Enhanced Functional Flow Block Diagram

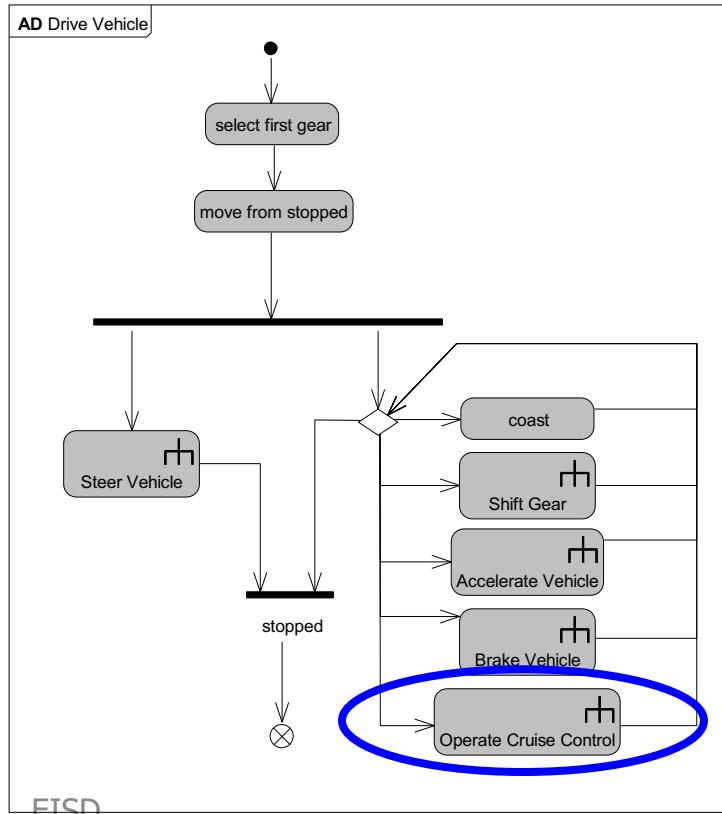
Analysis Model of Vehicle

- SysML additions on this chart
 - «streaming» activities consume inputs after initialization
 - «continuous» flows



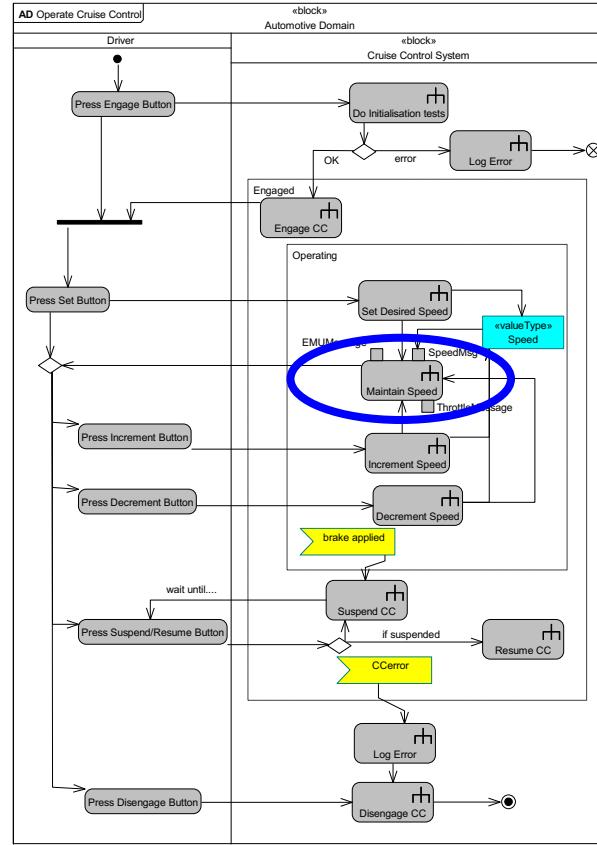
Analysis Model of Vehicle

- SysML additions on this chart
 - «streaming» activities consume inputs after initialization
 - «continuous» flows



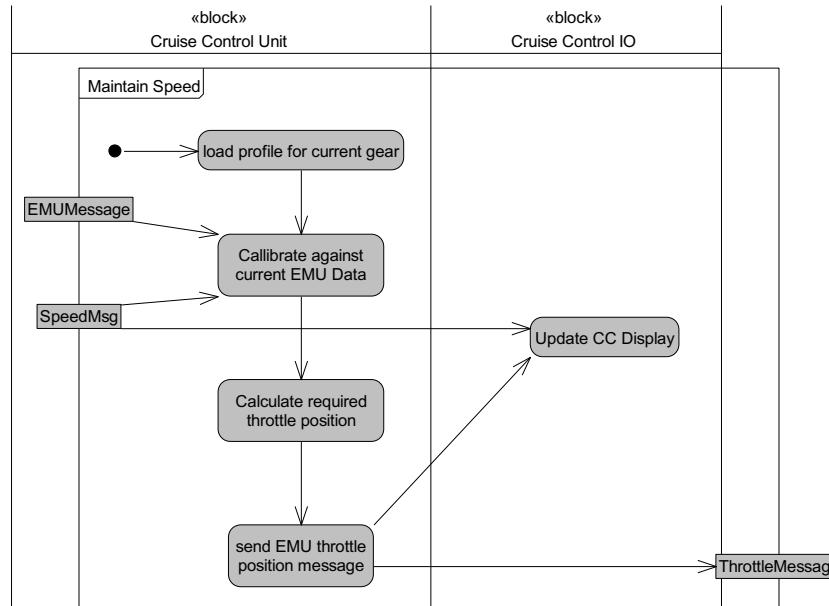
Analysis Model of Vehicle

- SysML additions on this chart
 - «streaming» activities consume inputs after initialization
 - «continuous» flows



Analysis Model of Vehicle

- SysML additions on this chart
 - «streaming» activities consume inputs after initialization
 - «continuous» flows



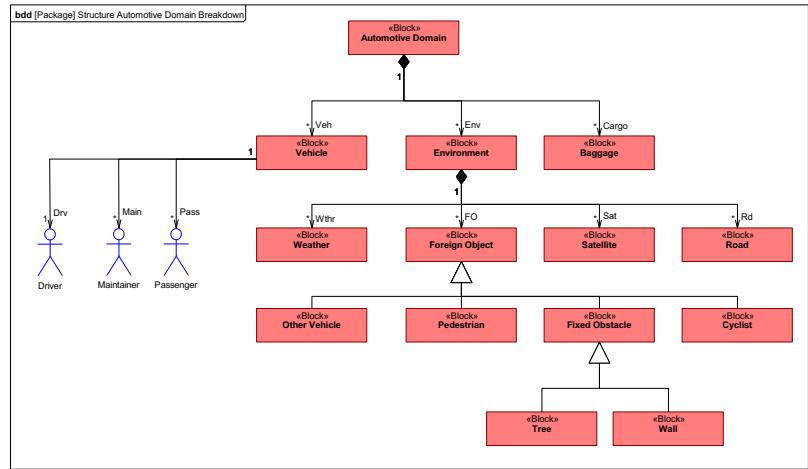
SysML Blocks

- Provides a unifying concept to describe the structure of an element or system
 - Hardware
 - Software
 - Data
 - Procedure
 - Facility
 - Person

Block is Basic Structural Element

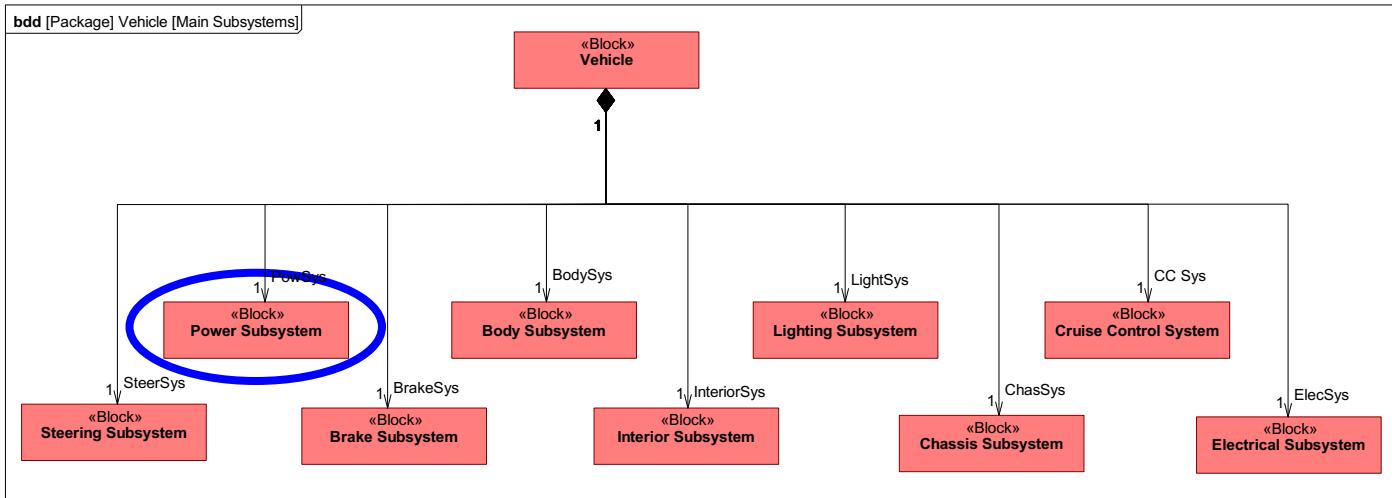
Block Definition Diagram

- Parts shown by black-diamond notation, or by Parts Compartment
- Values compartment shows properties of the block
- Flowports compartment shows block interface



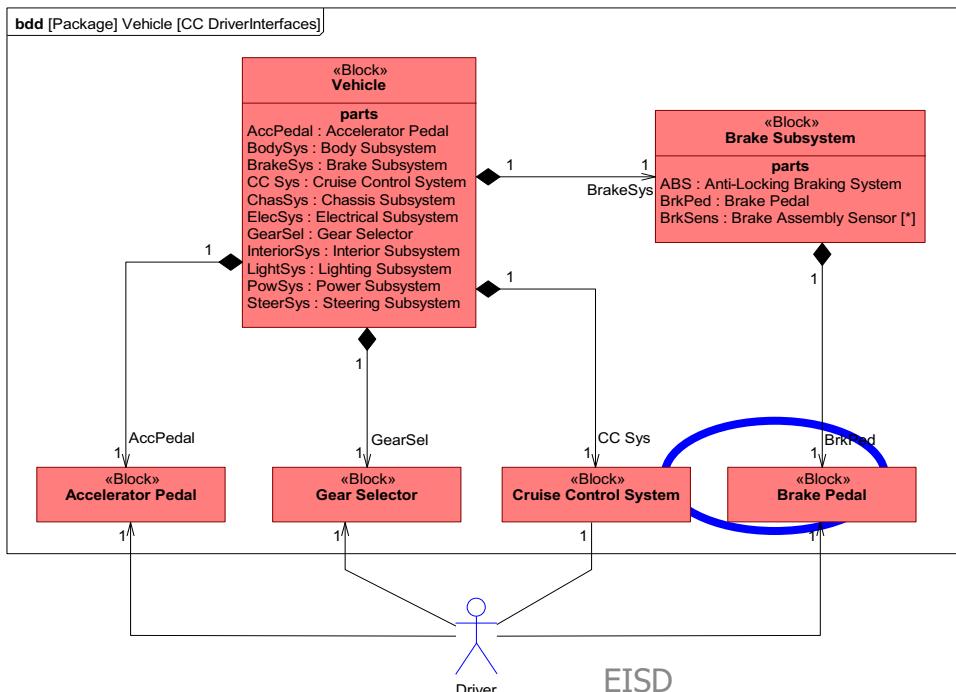
Block Definition Diagram for Vehicle Structural Components

- Parts shown by black-diamond notation, or by Parts Compartment
- Values compartment shows properties of the block
- Flowports compartment shows block interface



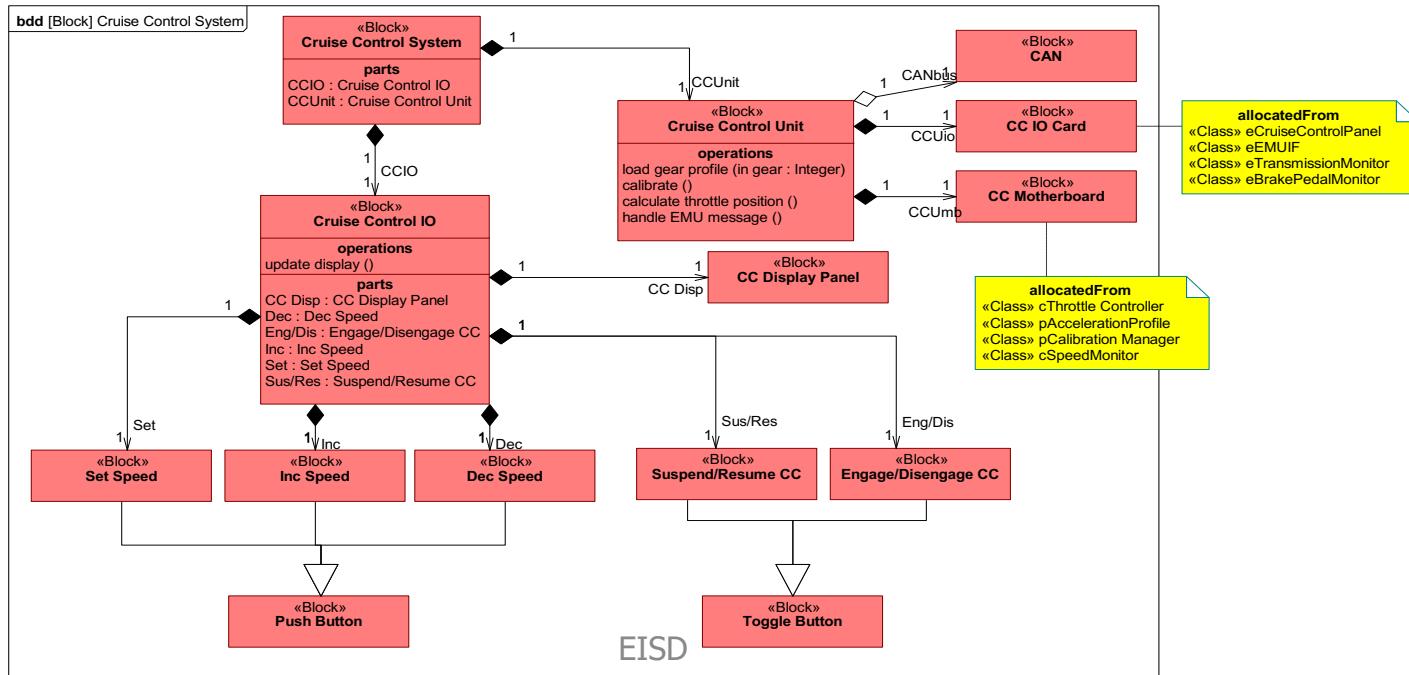
Block Definition Diagram for Vehicle Structural Components

- Parts shown by black-diamond notation, or by Parts Compartment
- Values compartment shows properties of the block
- Flowports compartment shows block interface



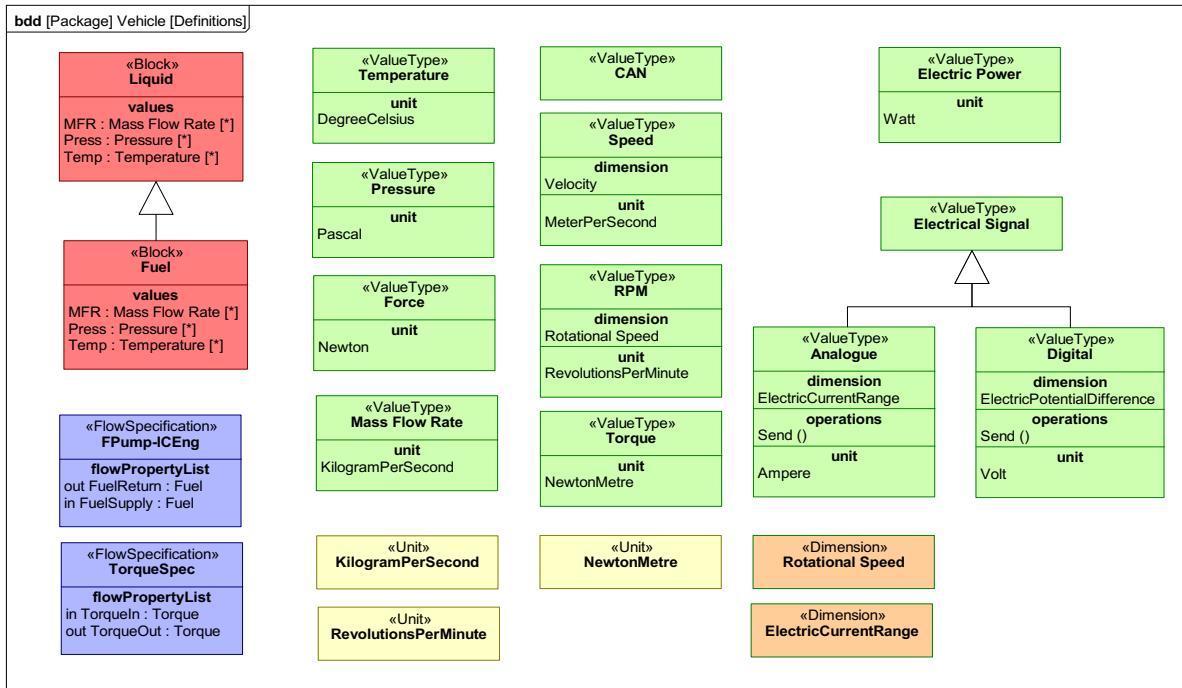
Block Definition Diagram for Vehicle Structural Components

- Parts shown by black-diamond notation, or by Parts Compartment
- Values compartment shows properties of the block
- Flowports compartment shows block interface



Unit and Item Types

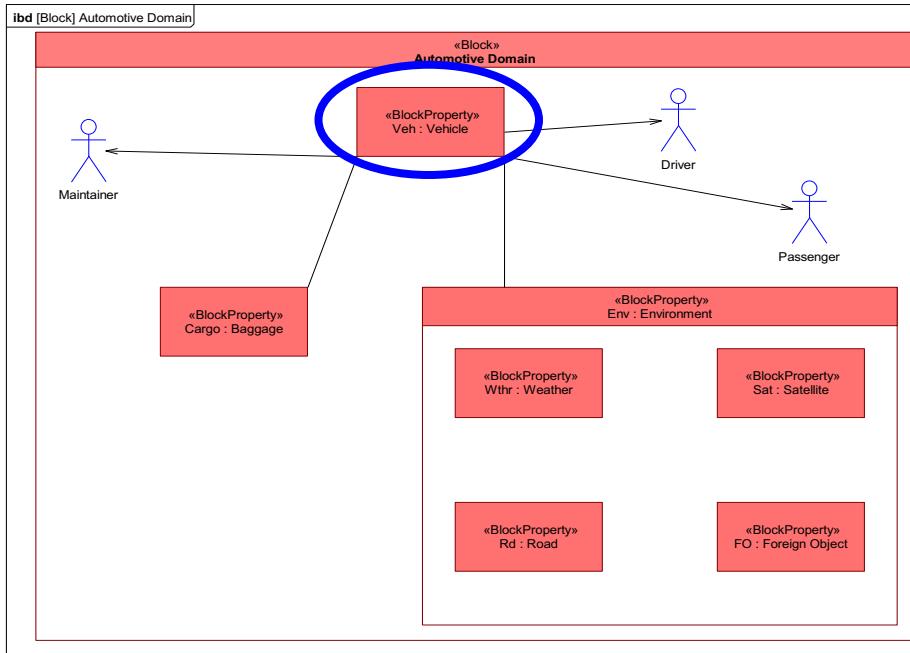
- Unit types normally based on Real
- SI Units and Dimensions defined in SysML appendix



Item Flows

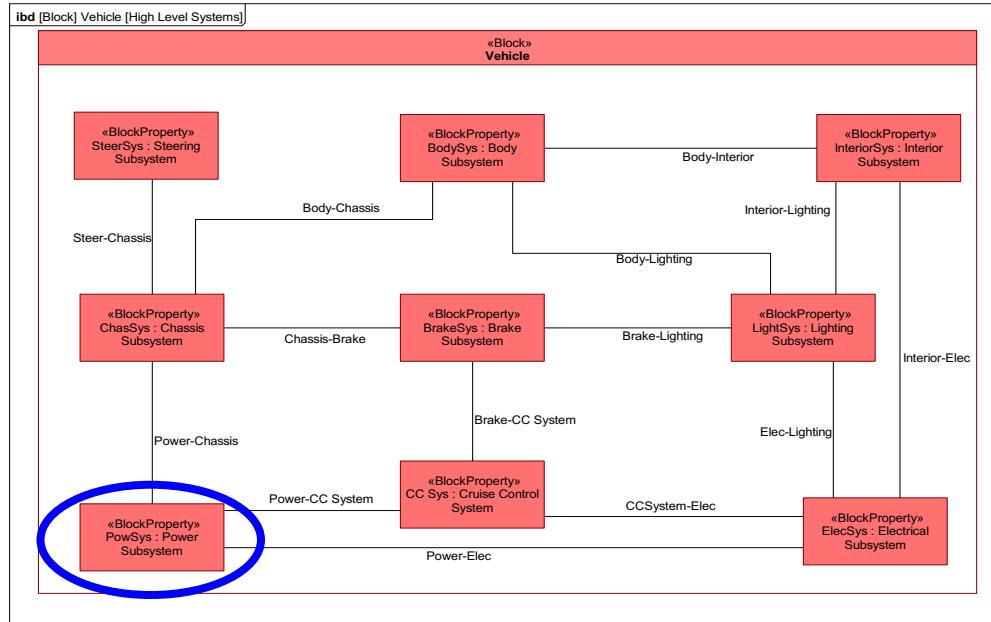
- Distinct from what can flow via the port specification
- Supports top down description of flows without imposing behavioural method (e.g. activities, state, interactions)
 - Behaviour is not driven from itemFlows but needs to be consistent with it
 - Is aligned with behaviour thru refinement and allocation
 - Can be allocated from an object node, message, or signal from a behavioural diagram
- Properties of item flow can be specified and constrained in parametric diagram

IBD for Vehicle



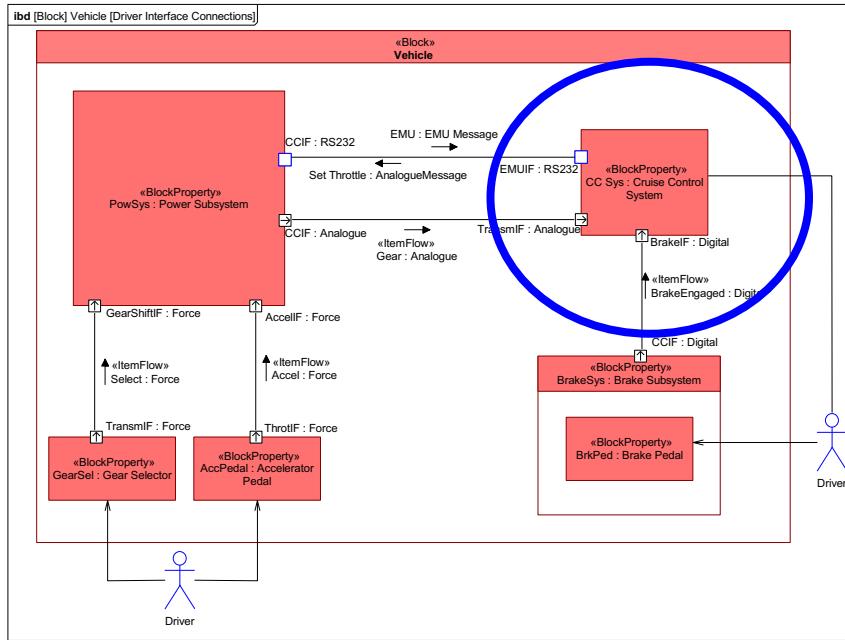
- Non-Atomic Ports
 - I/O is specified using FlowSpecification
 - FlowSpecification consists of properties stereotyped «FlowProperty»
 - isConjugate promotes reuse of flowSpecifications
- Atomic FlowPorts
 - In this case the port is directly typed by the item type (Block or ValueType)
 - Direction property specify the direction of flow

IBD for Vehicle



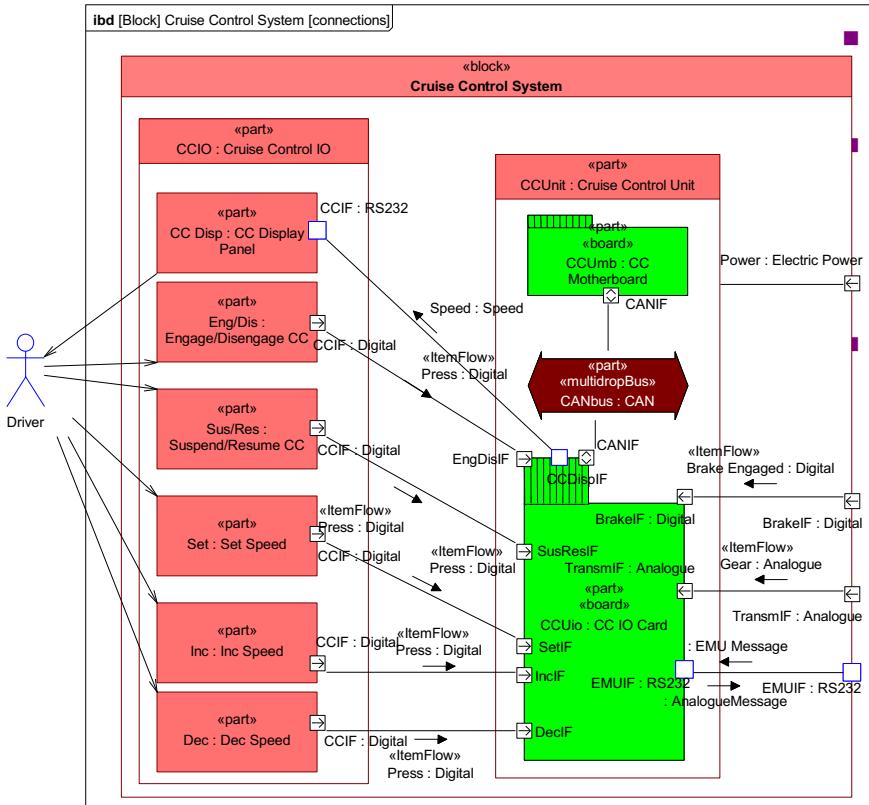
- Non-Atomic Ports
 - I/O is specified using FlowSpecification
 - FlowSpecification consists of properties stereotyped «FlowProperty»
 - isConjugate promotes reuse of flowSpecifications
- Atomic FlowPorts
 - In this case the port is directly typed by the item type (Block or ValueType)
 - Direction property specify the direction of flow

IBD for Vehicle



- Non-Atomic Ports
 - I/O is specified using FlowSpecification
 - FlowSpecification consists of properties stereotyped «FlowProperty»
 - isConjugate promotes reuse of flowSpecifications
- Atomic FlowPorts
 - In this case the port is directly typed by the item type (Block or ValueType)
 - Direction property specify the direction of flow

Internal Block Diagram for CC



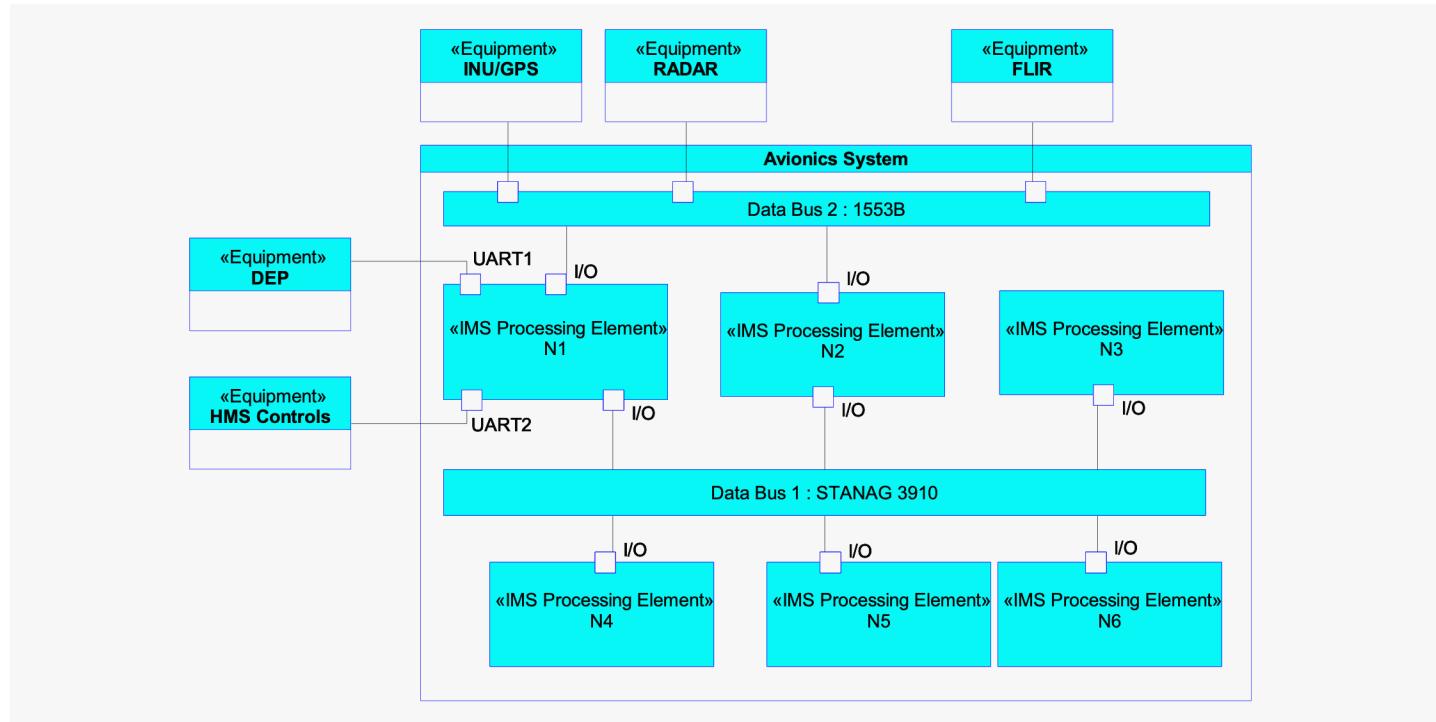
Shows parts (structural children) ...
... and ports (interaction points on blocks and parts)

- Supports integration of behavior and structure

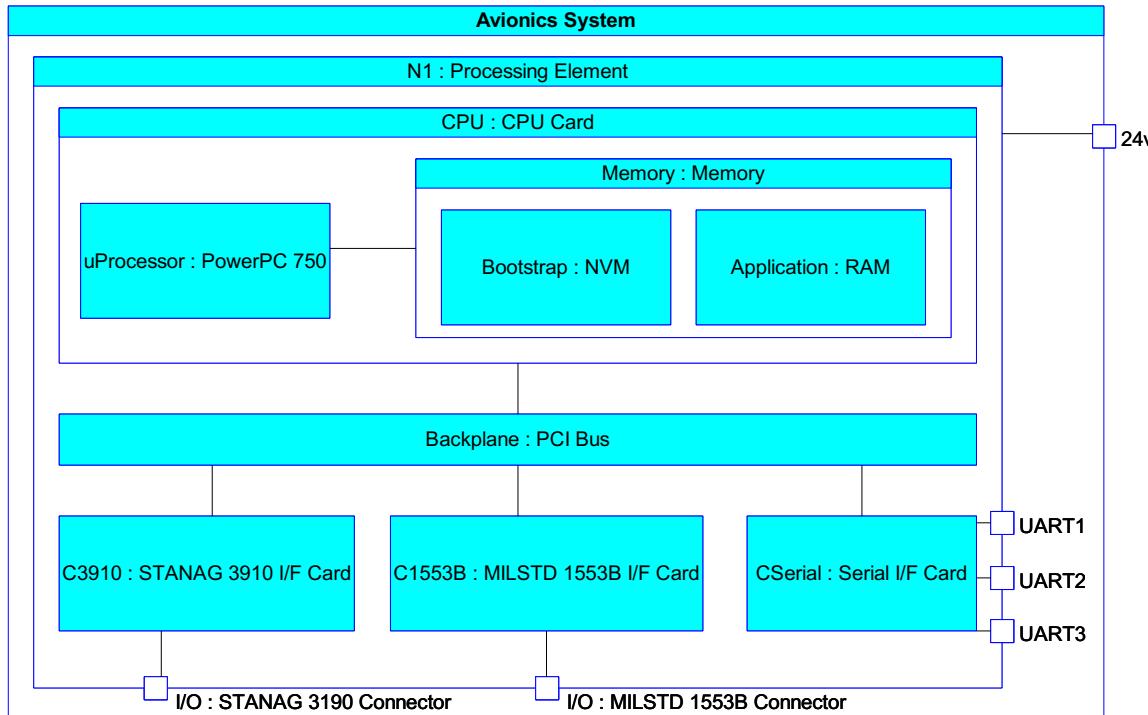
Port types

- Standard Ports
 - Specify a set of operations and/or signals
 - Typed by a UML interface
- Flow Ports
 - Specify what can flow in or out of block/part
 - Typed by a flow specification

Avionics - Topology of Processing Elements



Avionics - Internals of a Processing Element



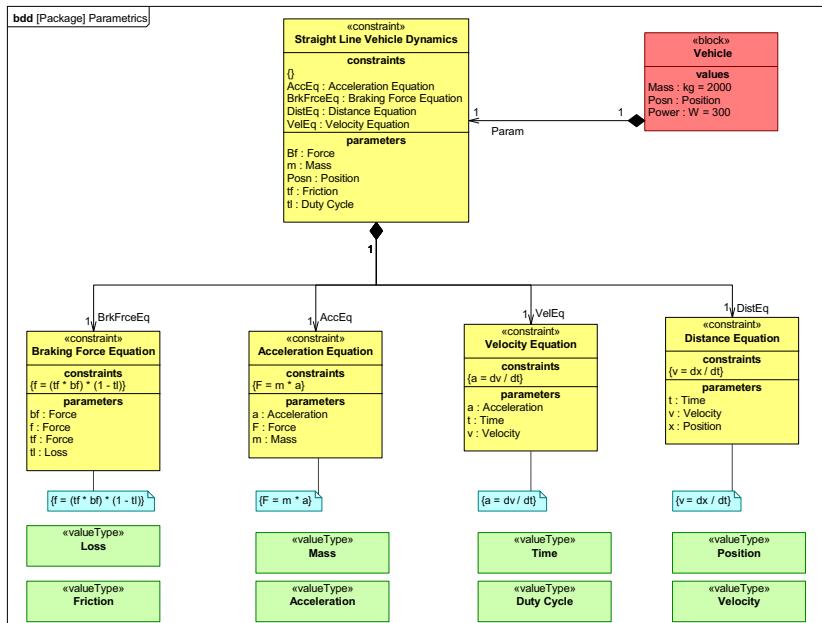
Parametrics

- Used to express constraints (equations) between value properties
 - Provides support to engineering analysis (e.g. performance, reliability, etc)
- Constraint block captures equations
 - Expression language can be formal (e.g. MathML, OCL ...) or informal
 - Computational engine is defined by applicable analysis tool and not by SysML
- Parametric diagram represents the usage of the constraints in an analysis context
 - Binding of constraint usage to value properties of blocks (e.g. vehicle mass bound to $F = m * a$)

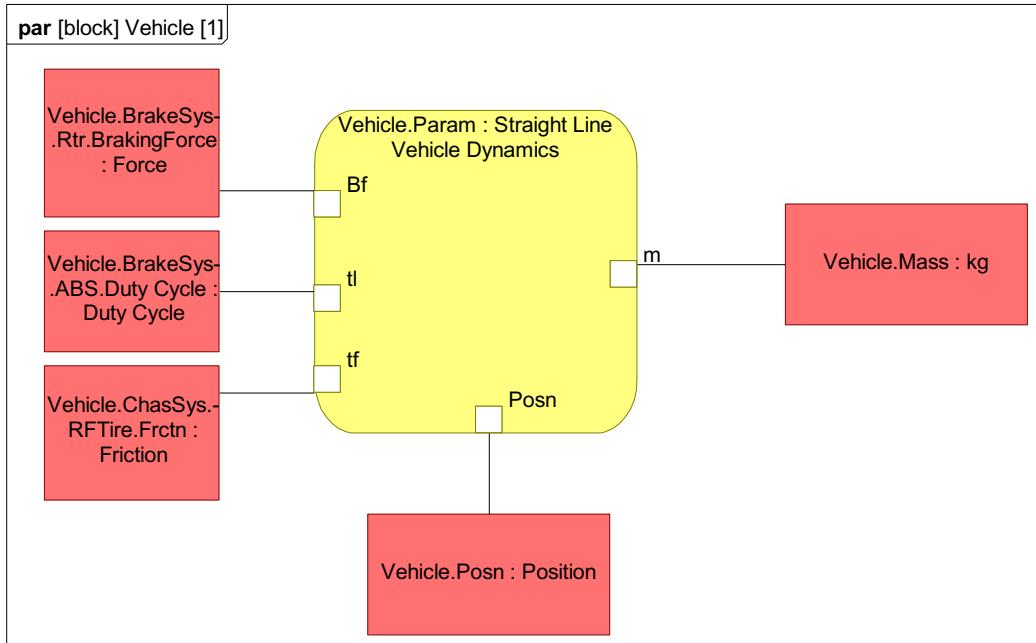
Parametrics Enable Integration of Engineering Analysis with Design Models

Vehicle Parametrics BDD

- BDDs can show parametric definitions
- Parameter compartment shows the constraint parameters
- Constraint compartment for the applied constraint

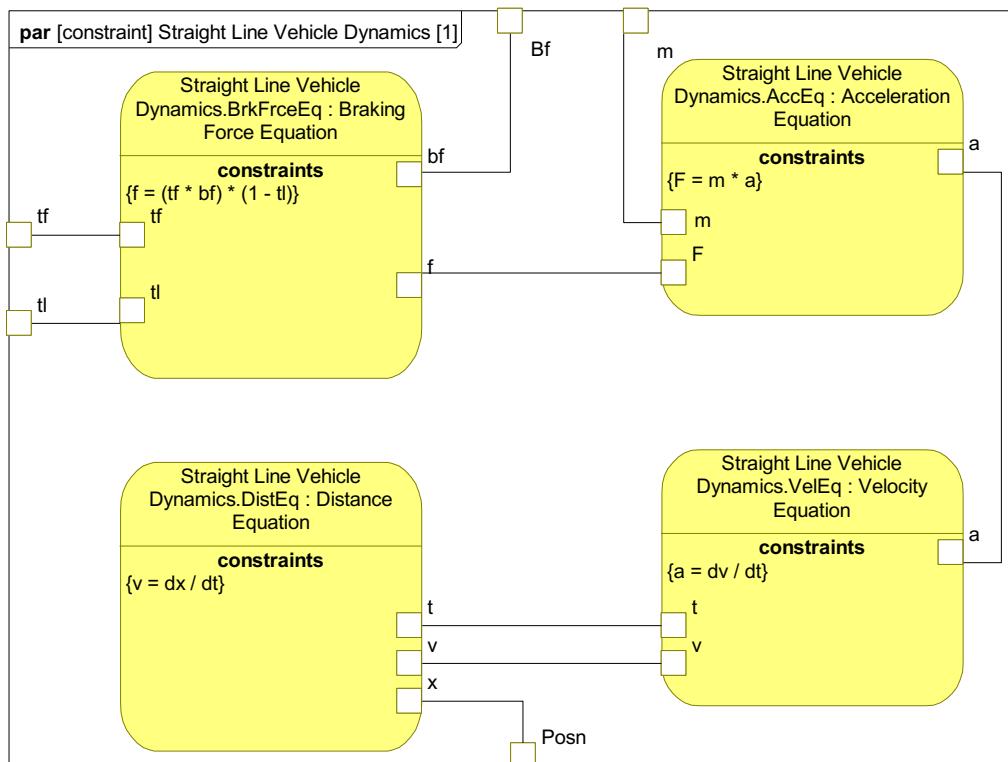


Parametric Diagram



- Small boxes represent parameters and bound properties
- Boxes on left represent item flows
- Constraint can be in compartment or in attached note

Parametric Diagram

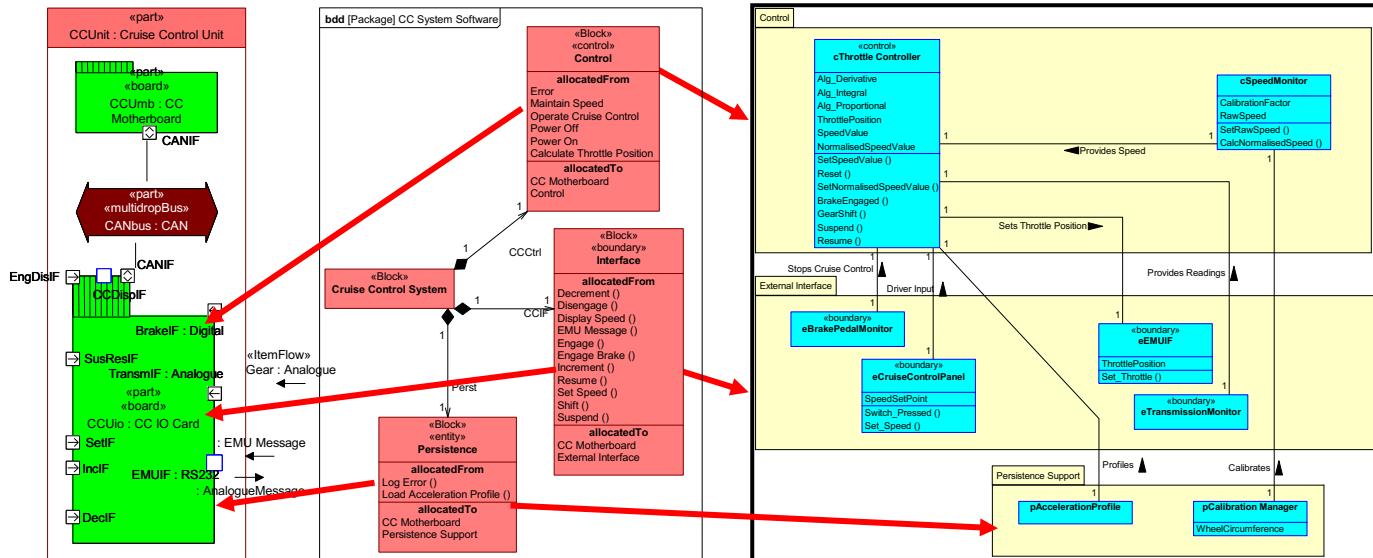


- Small boxes represent parameters and bound properties
- Boxes on left represent item flows
- Constraint can be in compartment or in attached note

Allocations

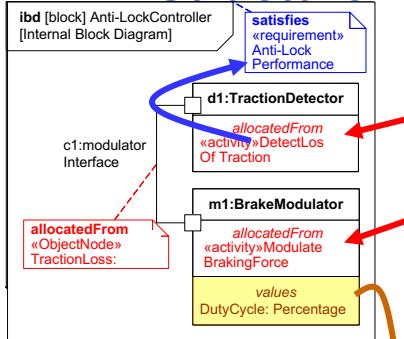
- Provides general relationship to map one model element to another
- Different types of allocation may include:
 - Behavioural (i.e. function to component)
 - Structural (i.e. logical to physical)
 - Hardware to Software
 -
- Explicit allocation of activities to swim lanes (e.g. activity partitions)
- Use of graphical and/or tabular representations

Allocation to SW/HW

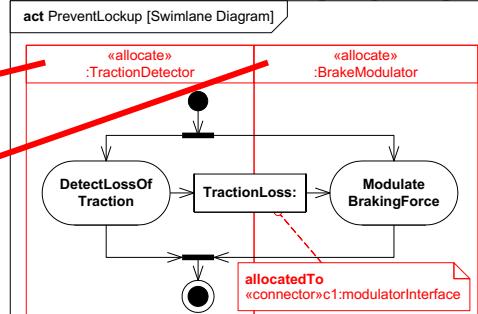


Cross Connecting Model Elements

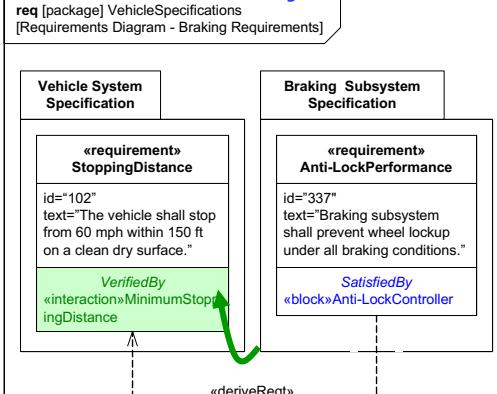
1. Structure



2. Behavior



satisfy

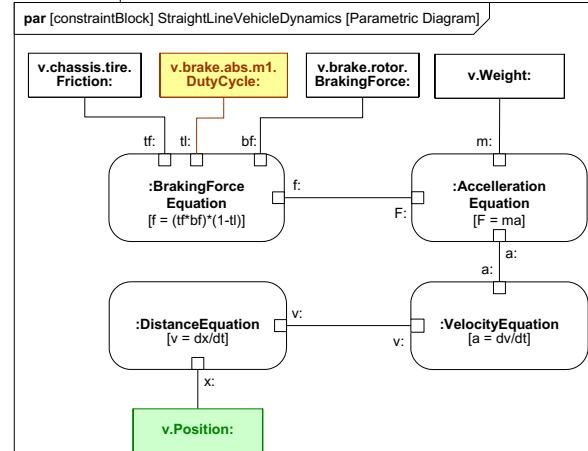


3. Requirements

verify EISD

verify EISD

4. Parametrics



Summary

- SysML sponsored by INCOSE/OMG with broad industry and vendor participation
- SysML provides a general purpose modelling language to support specification, analysis, design and verification of complex systems
 - Subset of UML 2 with extensions
 - 4 Pillars of SysML include modelling of requirements, behaviour, structure, and parametrics
- OMG SysML Adopted in May 2006
- Multiple vendor implementations announced
- Standards based modelling approach for SE expected to improve communications, tool interoperability, and design quality

Si arriva ad un punto in cui i blocchi sono pronti per essere trasformati in SW o HW