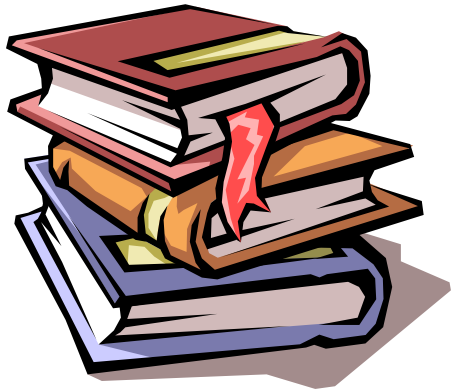
Systems Modeling Language (SysML) Overview

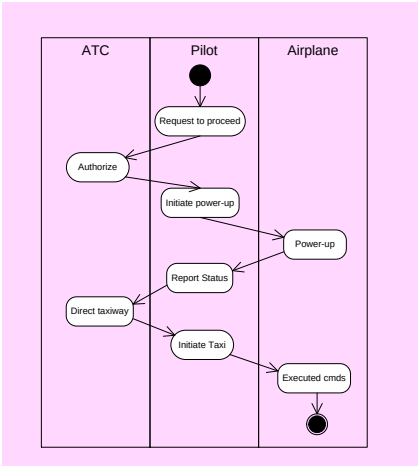
SE Practices for Describing Systems

Past



- Specifications
- Interface requirements
- System design
- Analysis & Trade-off
- Test plans

Future



Moving from Document centric to Model centric

Why Model Based Approach ?

- Improved communications
- Reduced ambiguity
- Reduced errors
- More complete representation
- Enhanced knowledge capture

What is SysML?

- SysML: Systems Modeling Language
- Aimed at systems engineering applications
  - It supports specification, analysis, design, verification and validation
- Enhances the ability to manage growing system complexity
- Initially intended as an open source project
- An extension to UML

What is SysML?

- SysML is a language, not a tool
- It is supported by a number of modelling tools, for example:
  - Topcased (open-source, Eclipse-based)
  - Papyrus (open-source, Eclipse-based)
  - IBM Rational (commercial)
- Concept of model repository: the modelling tool stores the model in a repository
  - Each SysML diagram is just a particular view of the model repository

Unified Modeling Language (UML)

- UML
  - Is a visual modeling language
  - Is not a methodology
  - Defacto standard for SW level OO modeling

- Visual Modeling Language = Notation + Semantics
  - Semantics = meaning
  - Notation = representation of meaning

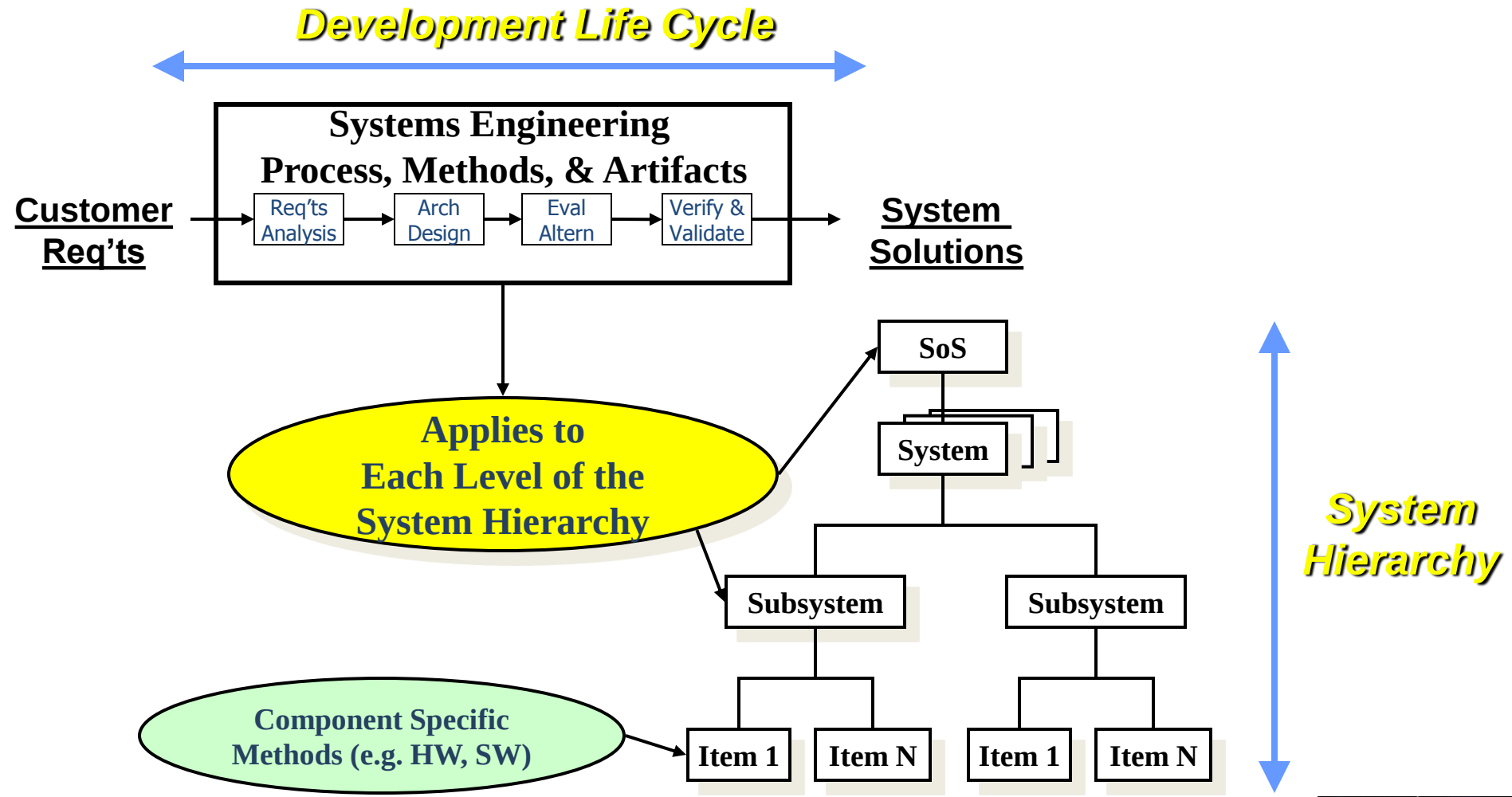
UML/SysML Status

- UML V2.0
  - Updated version of UML that offers significant capability for systems engineering over previous versions
  - Adopted in June 2003/Finalized in 2005

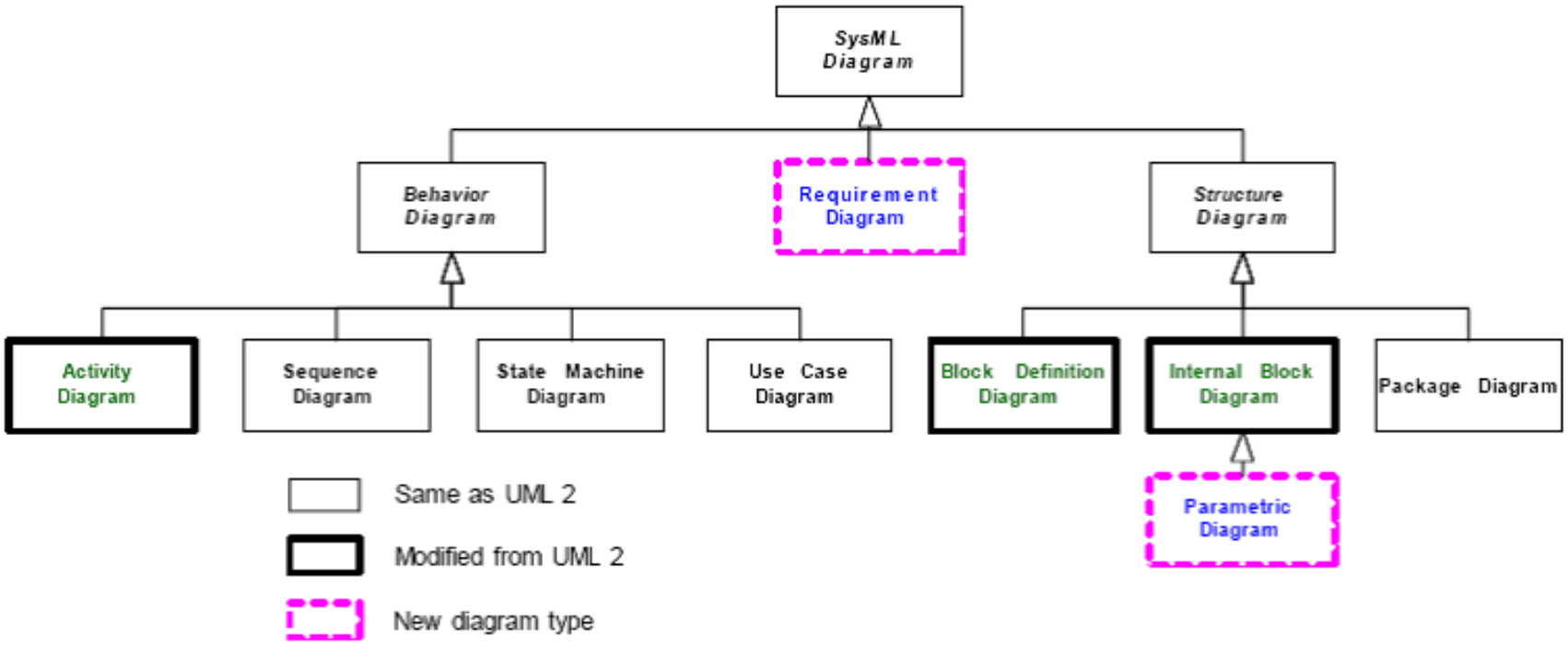
- UML for Systems Engineering (SE) RFP
  - Established the requirements for a system modeling language
  - Issued by the OMG in March 2003

- SysML
  - Industry Response to the UML for SE RFP
  - Addresses most of the requirements in the RFP
  - V1.0 adopted in 2005

SysML Scope


Development Life Cycle
Systems Engineering
Process, Methods, & Artifacts
Req'ts Analysis
Arch Design
Eval Altern
Verify & Validate
Customer Req'ts
System Solutions
Applies to Each Level of the System Hierarchy
SoS
System
Subsystem
Subsystem
Item 1
Item N
Item 1
Item N
Component Specific Methods (e.g. HW, SW)
System Hierarchy

SysML Diagram Taxonomy



SysML Drawing Summary

SYSML DIAGRAM	PURPOSE	UML ANALOG
Activity diagram	Show system behavior as control and data flows. Useful for functional analysis. Compare Extended Functional Flow Block diagrams (EFFBDs), already commonly used among systems engineers.	**Activity diagram**
Block Definition diagram	Show system structure as components along with their properties, operations and relationships. Useful for system analysis and design.	**Class diagram**
Internal Block diagram	Show the internal structures of components, including their parts and connectors. Useful for system analysis and design.	**Composite Structure diagram**
Package diagram	Show how a model is organized into packages, views and viewpoints. Useful for model management.	**Package diagram**
Parametric diagram	Show parametric constraints between structural elements. Useful for performance and quantitative analysis.	N/A
Requirement diagram	Show system requirements and their relationships with other elements. Useful for requirements engineering.	N/A

SysML Drawing Summary - cont.

Sequence diagram	Show system behavior as interactions between system components. Useful for system analysis and design.	**Sequence diagram**
State Machine diagram	Show system behavior as sequences of states that a component or interaction experience in response to events. Useful for system design and simulation/code generation.	**State Machine diagram**
Use Case diagram	Show system functional requirements as transactions that are meaningful to system users. Useful for specifying functional requirements. (Note potential overlap with Requirement diagrams.)	**Use Case diagram**
Allocation tables* *dynamically derived tables, not really a diagram type	Show various kinds of allocations (e.g., requirement allocation, functional allocation, structural allocation). Useful for facilitating automated verification and validation (V&V) and gap analysis.	N/A

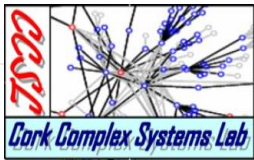
UML 2.0 Component, Communications, Object, Deployment, Interaction and Timing diagrams have not been included in SysML

4 Pillars of SysML

- OMG SysML includes diagrams that can be used to specify:
  - Requirements
  - Structure
  - Behaviour
  - Parametric Relationships (Parametrics)

4 Pillars of SysML

Requirements



Requirements

Requirement Diagram

- Collection of system requirements organized hierarchically

- Requirement may specify
  - a function that a system must perform (functional requirements), or
  - a performance condition that a system must satisfy (non-functional requirements).

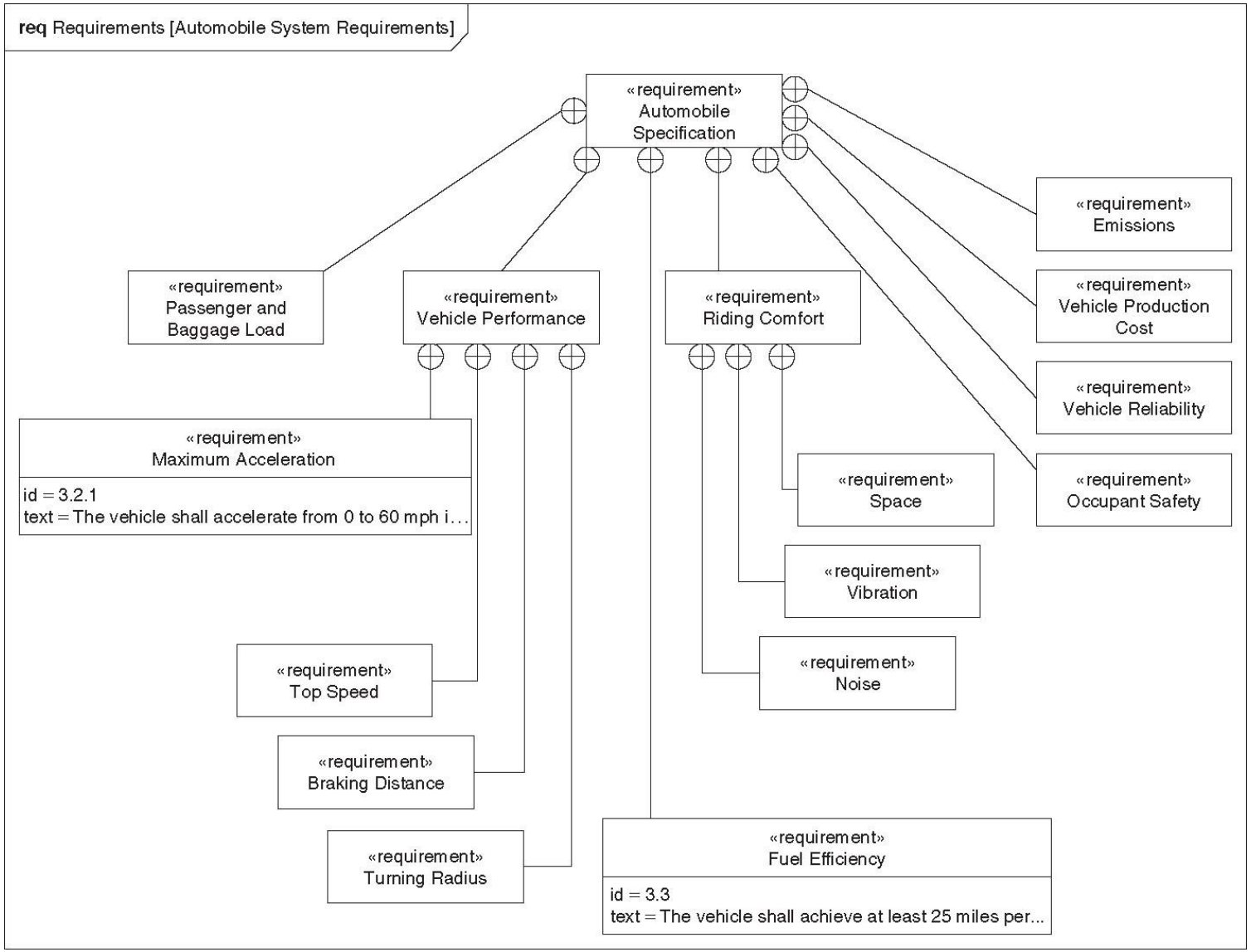
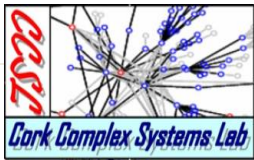


Figure 3.2

4 Pillars of SysML

Structure



Structural Constructs

Block Definition Diagram

- Represents blocks in terms of their features and their relationships with other blocks

- Block = an entity, a structural unit with one or more distinguishing features

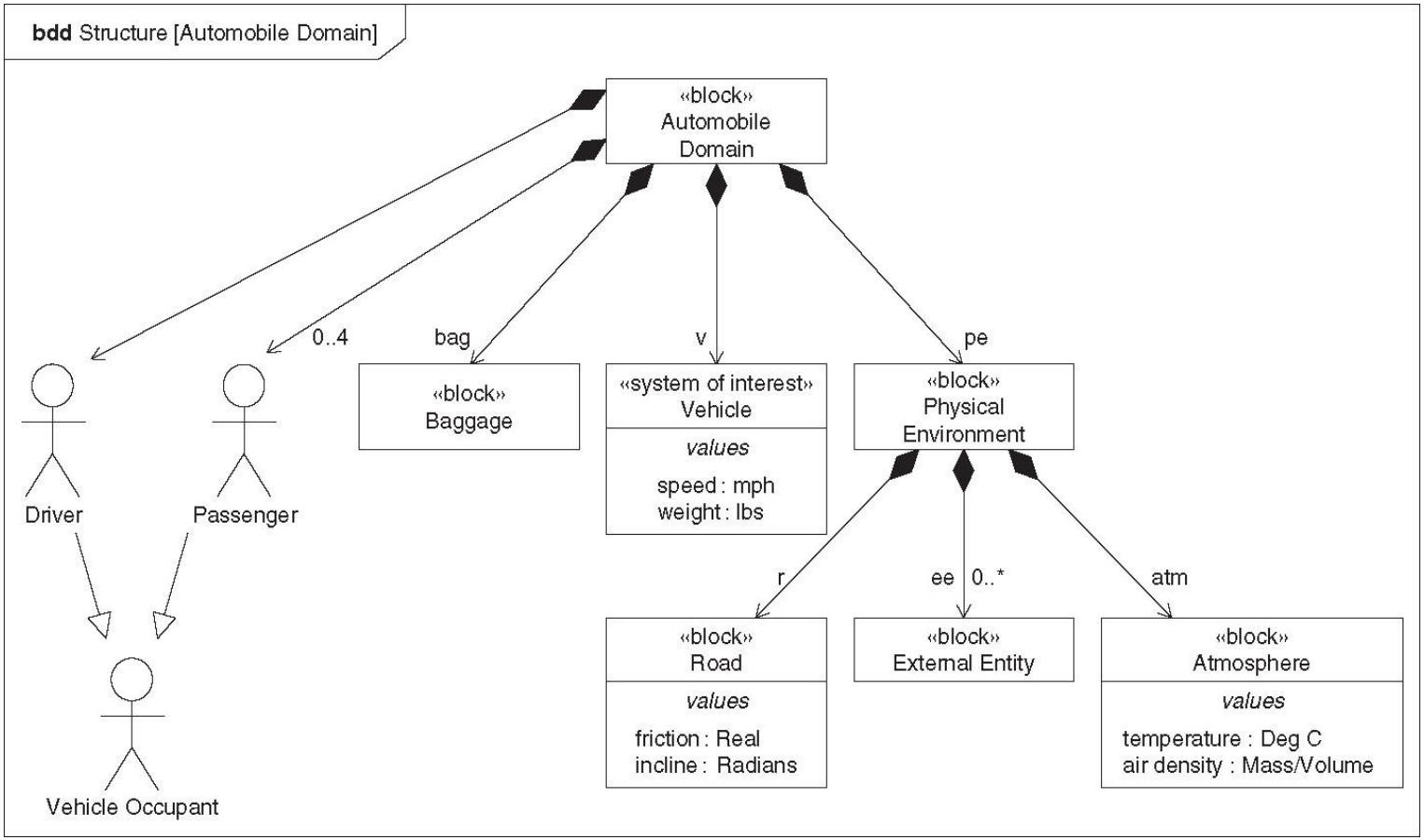


Figure 3.3

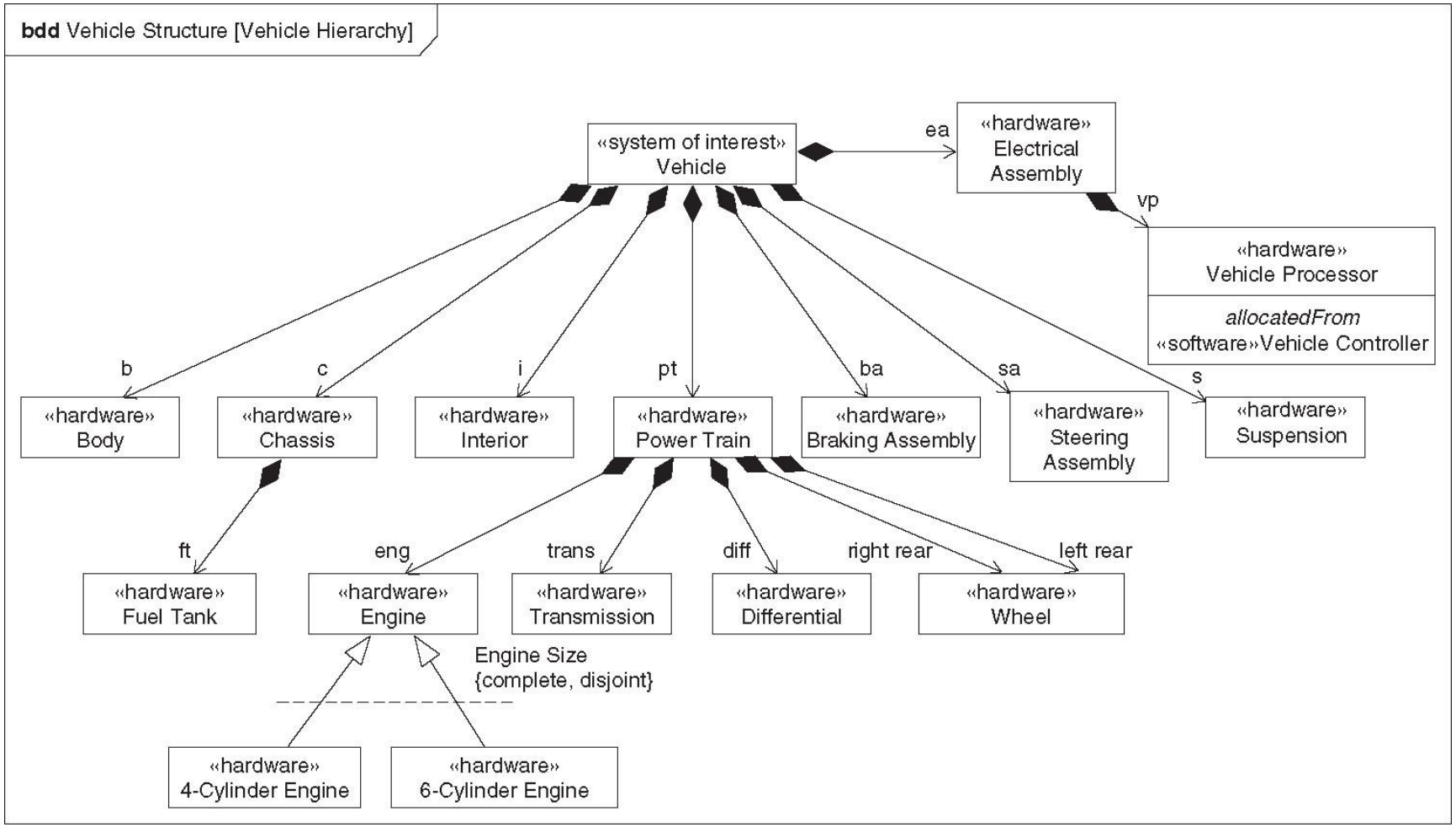


Figure 3.10

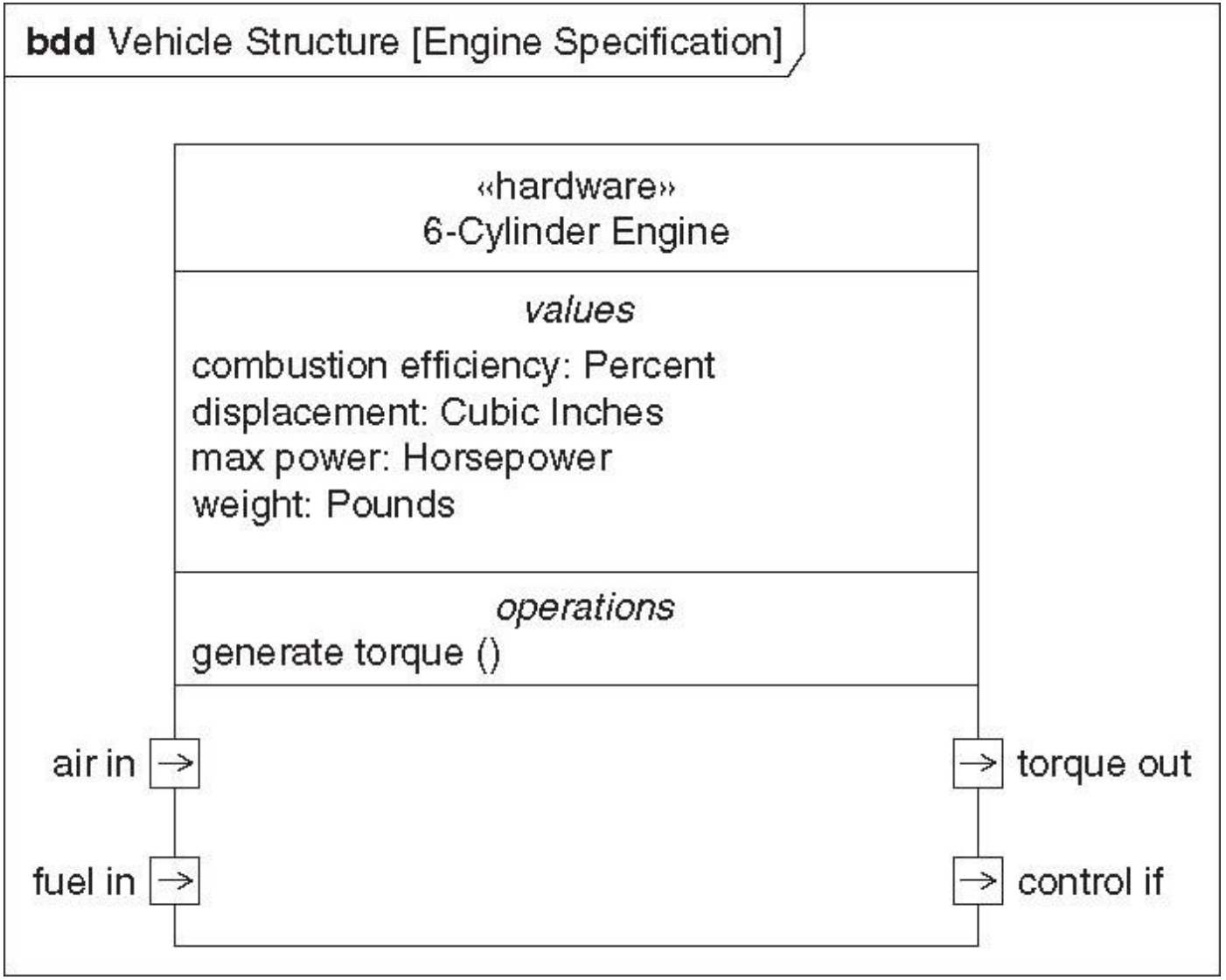


Figure 3.17

Structural Constructs

Internal Block Diagram

- Shows the internal structure of a block

- Shows how parts are connected and how they exchange information

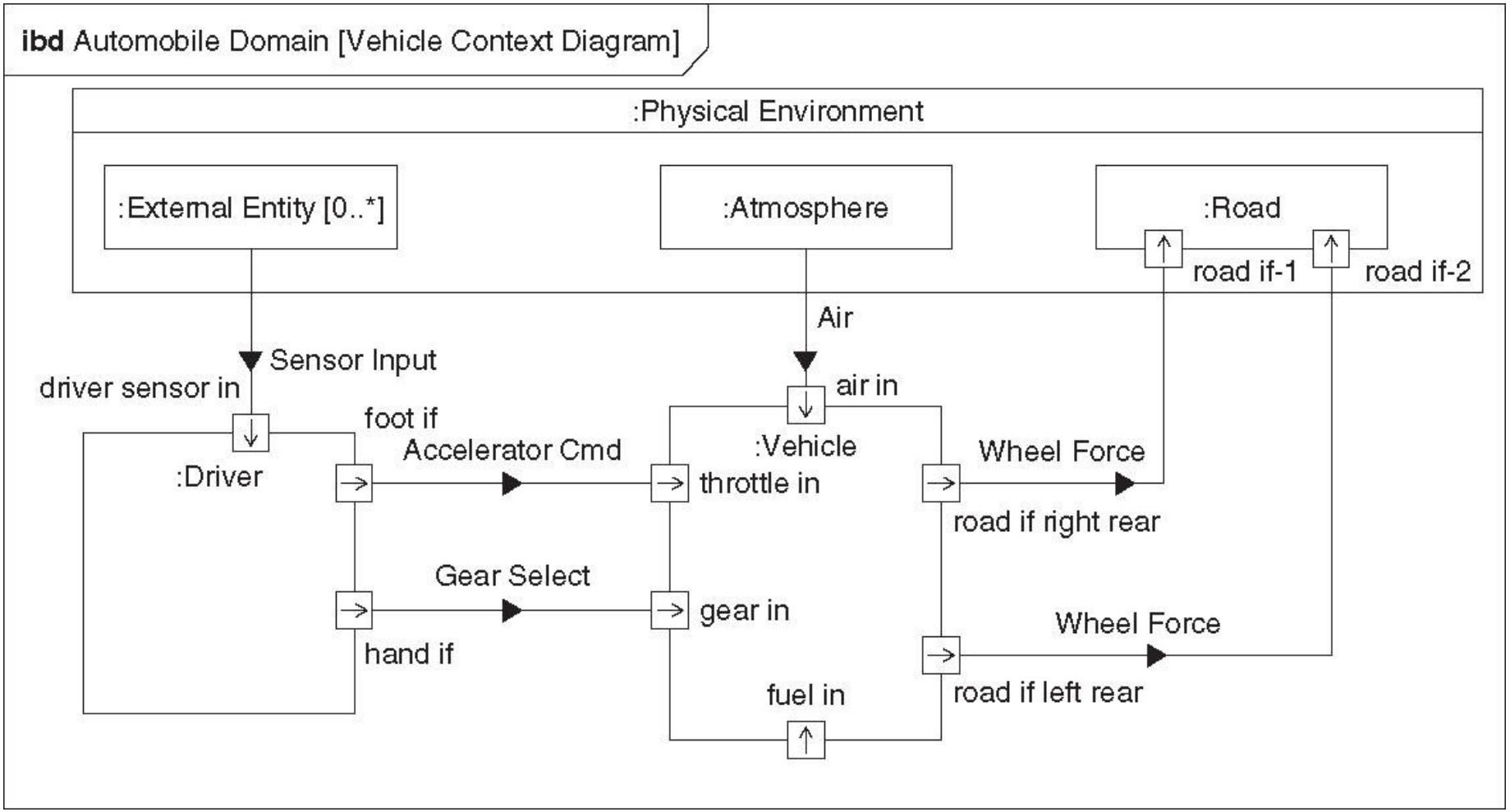
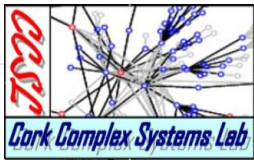


Figure 3.9

4 Pillars of SysML

Behaviour



Behavioral Constructs

Use Case Diagram

- Remains unchanged (same as UML 2.0)
- Describes the interaction between an actor and a subject to achieve a goal

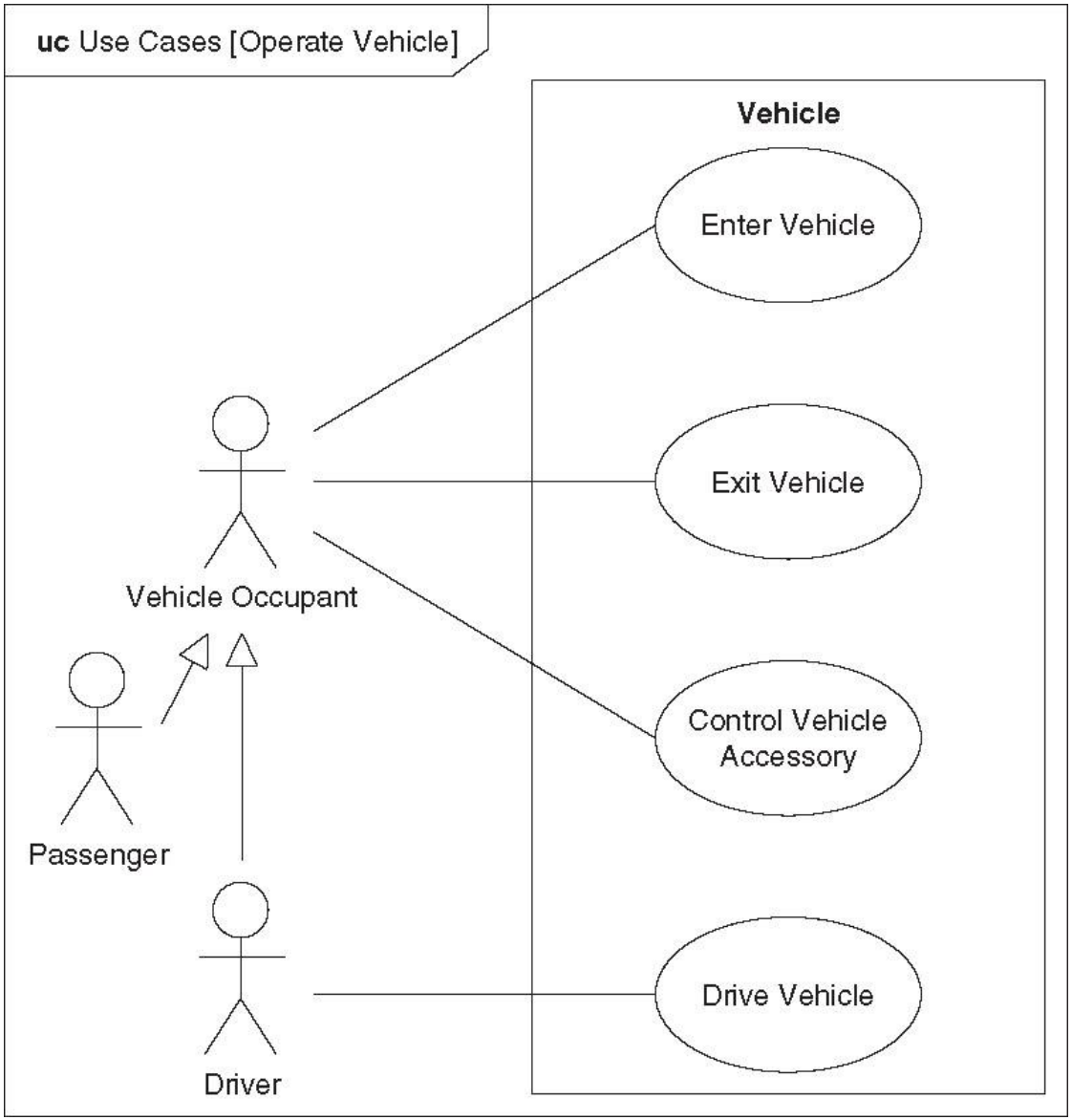


Figure 3.4

Behavioral Constructs

Sequence Diagram

- Remains unchanged (same as UML 2.0)

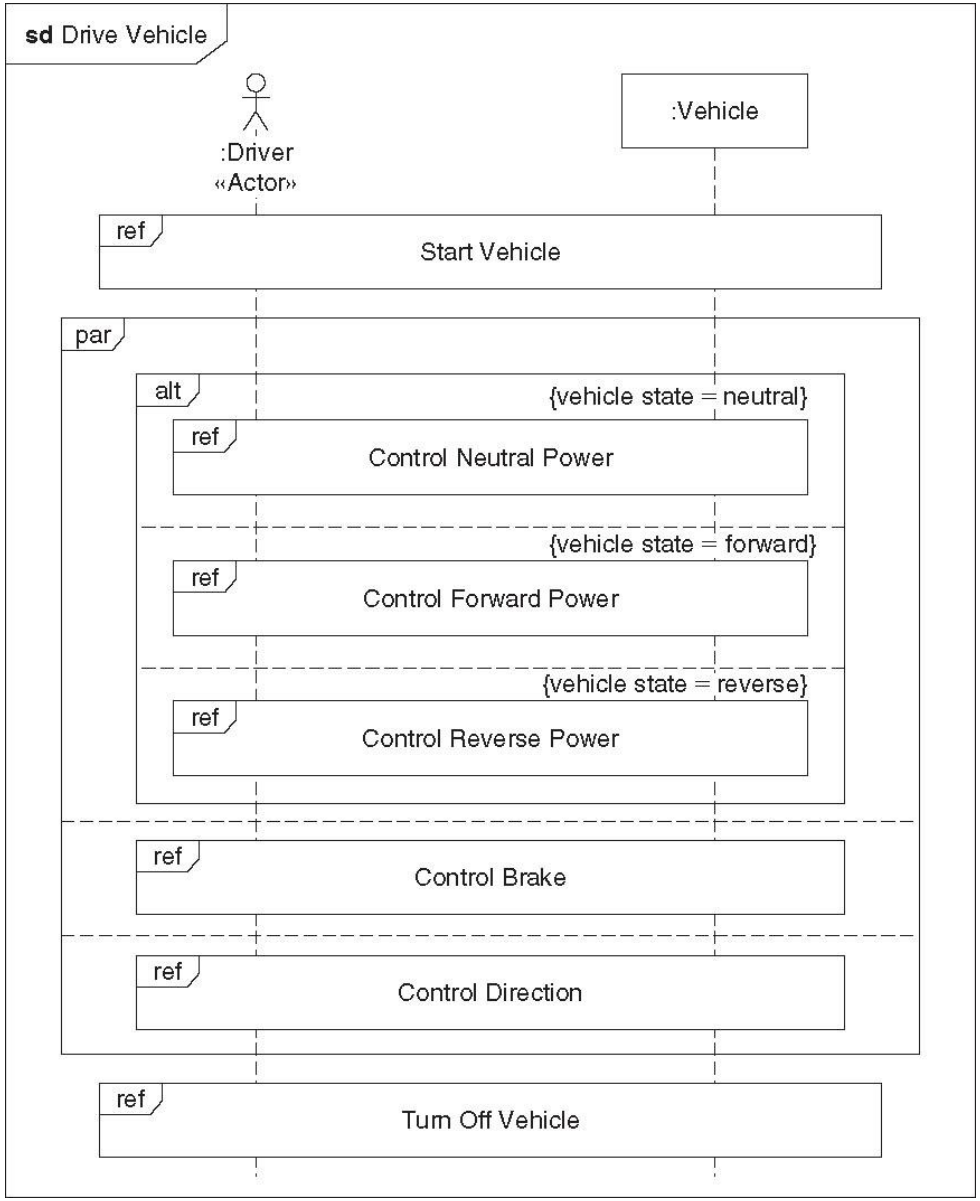


Figure 3.5

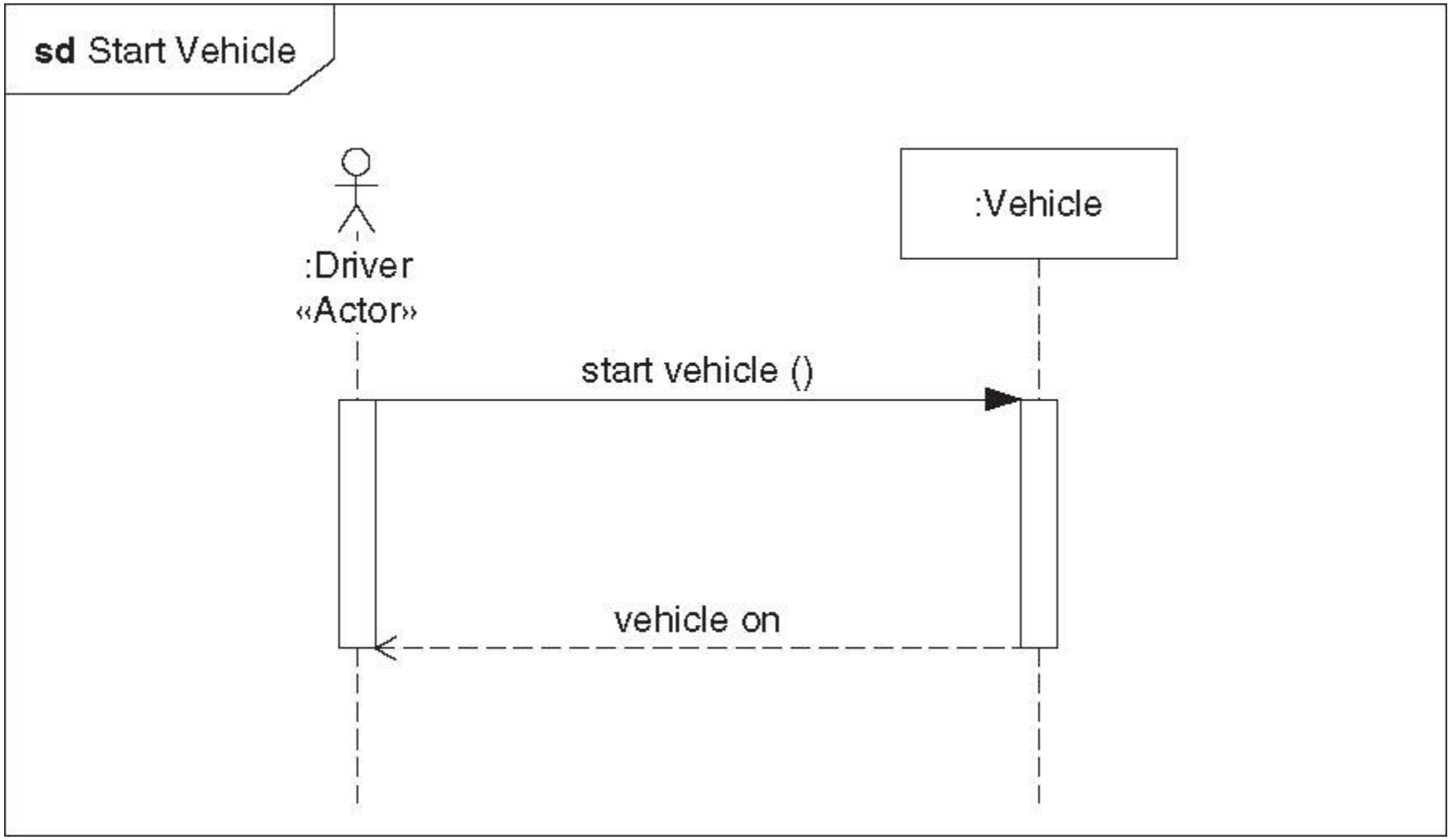


Figure 3.6

Behavioral Constructs

Activity Diagram

Extensions to UML 2

1. **Control as Data** –
   - Control can stop the actions that are executing.
2. **Continuous systems** –
   - Any sort of distributed flow of information & physical items
     through system (also includes discrete systems)
3. **Probabilities of flow**

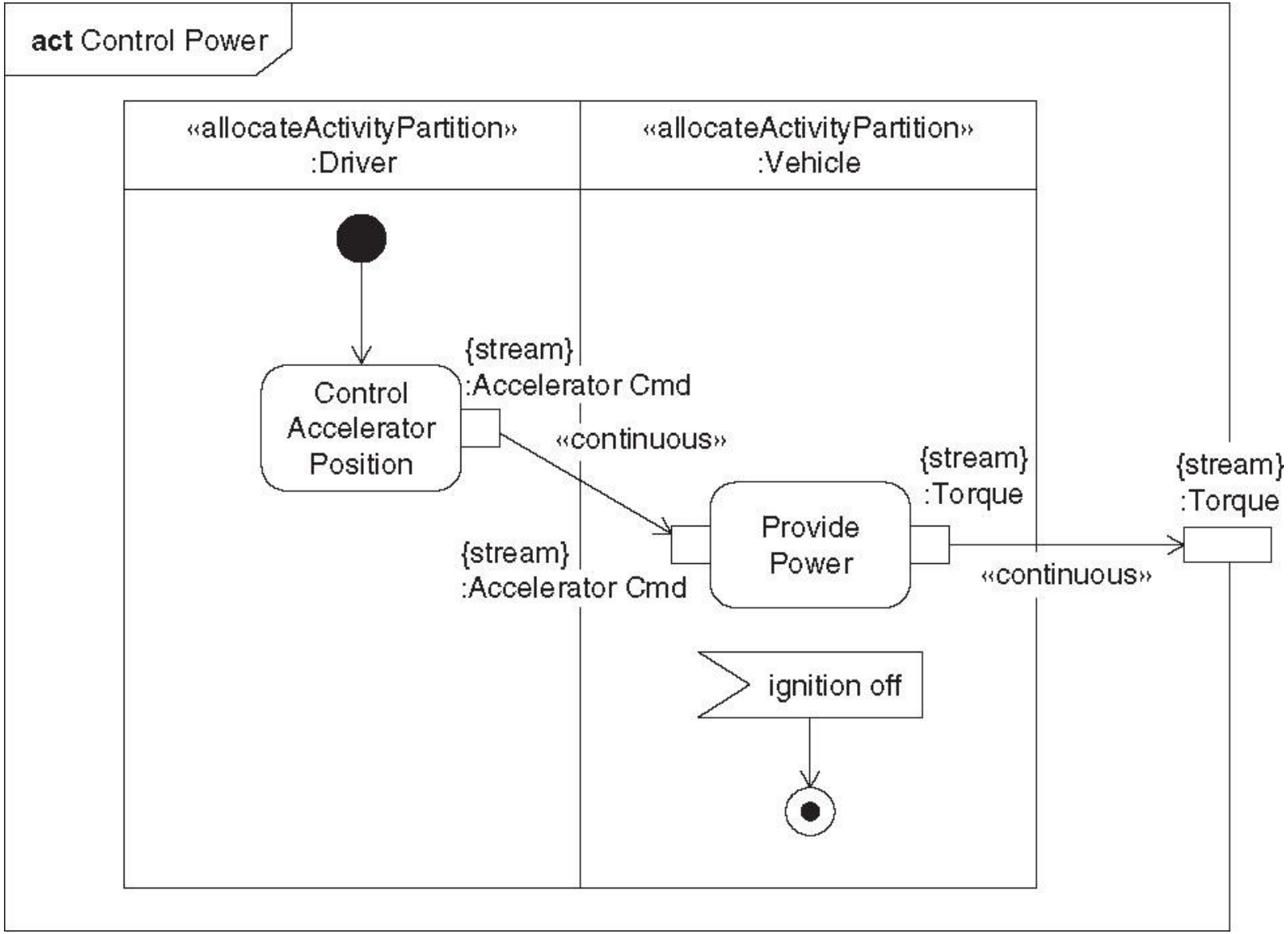
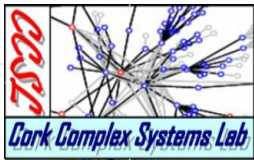


Figure 3.7

4 Pillars of SysML

Parametrics



Structural Constructs

Parametric Diagram

- Parametric models are useful to analyze the performance of a system.

- We start with blocks (constraintBlocks) that can contain constraints in terms of equations and parameters.

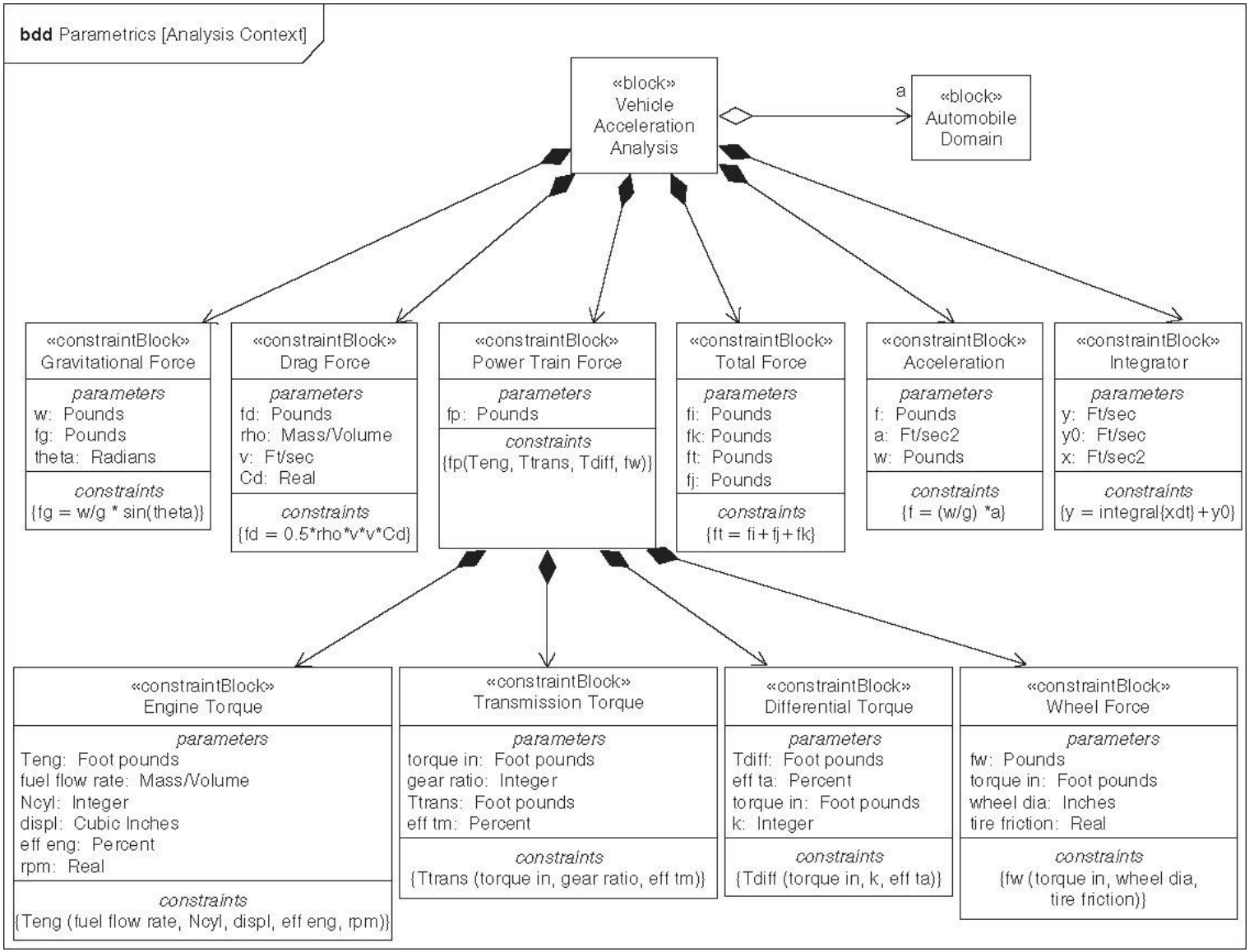


Figure 3.13

Structural Constructs

Parametric Diagram

- The Parametric Diagram specifies how constraints (equations) are connected.

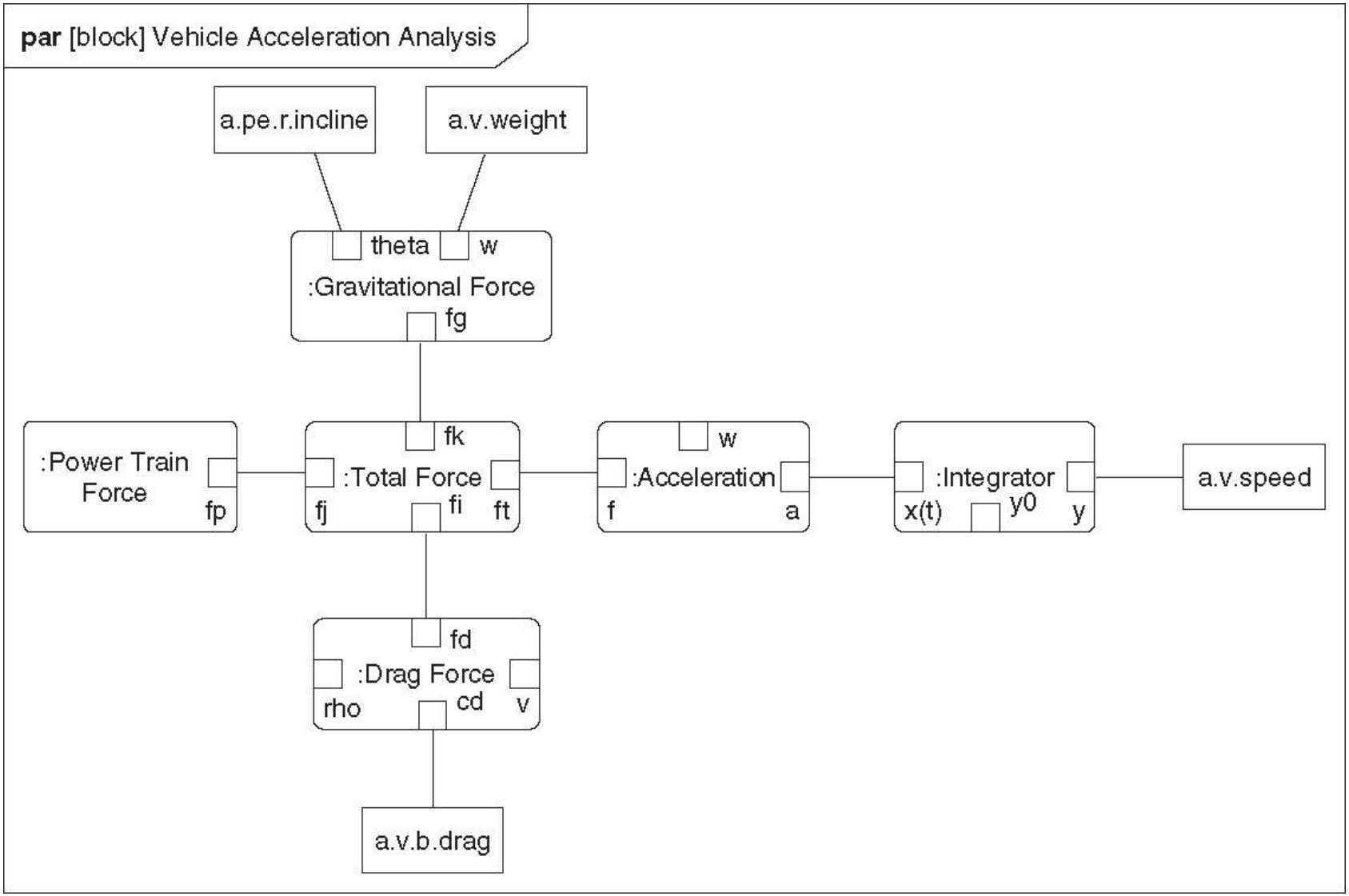


Figure 3.14

Structural Constructs

Parametric Diagram

- The Parametric Diagram can be provided to a simulation/analysis tool to execute test cases (normally separated from the SysML modeling tool)

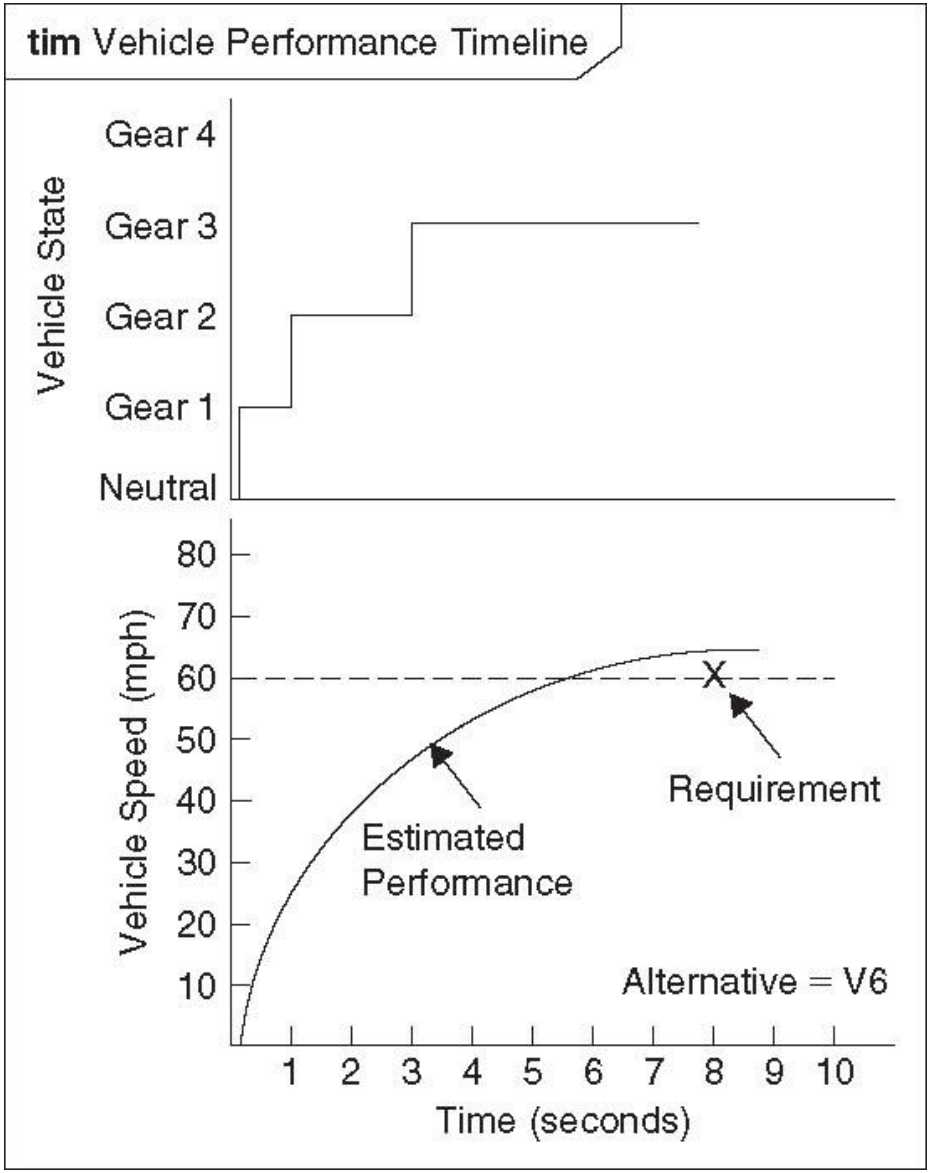


Figure 3.15

Cross Cutting Constructs

Requirement Diagram

- Requirement traceability: does the system satisfy the initial requirements?

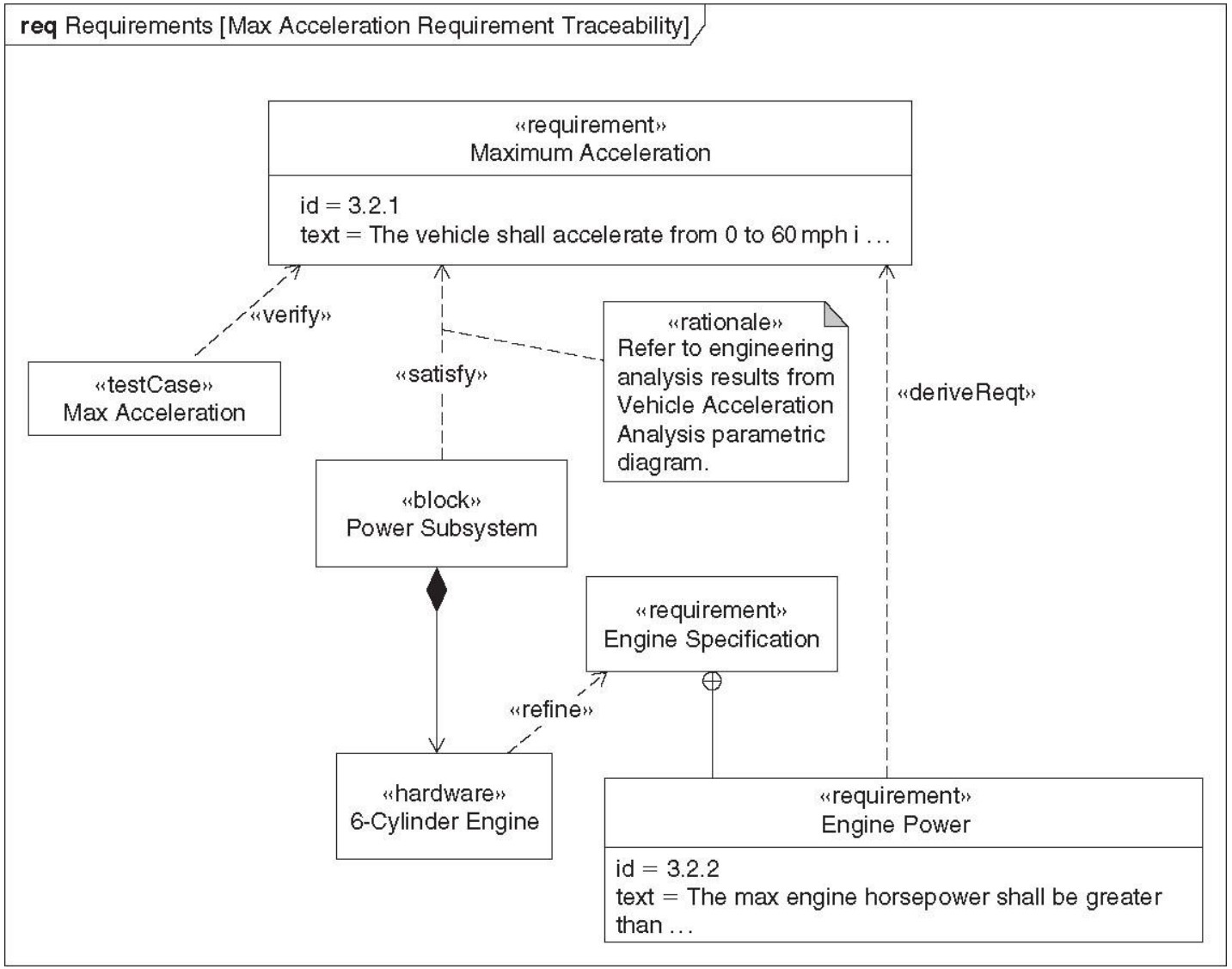


Figure 3.18

Summary

- SysML is the language for Systems Engineering
- Developed in Response to the need of standardizing systems engineering designs and process
- An extension to UML
- 2 new diagrams introduced
  - Requirement
  - Parametric

References

- http://www.sysmlforum.com/FAQ.htm#What_is_the_relationship_between_SysML_and_OMG_SysML
- http://www.omgsysml.org/
- http://en.wikipedia.org/wiki/SysML#Model_Exchange
- http://www.ibm.com/developerworks/rational/library/aug06/balmelli/
- http://www.brass-bullet.co.uk/view/global/page/94287/block/content/file/attachments/IET_SysMl_sept06.pdf
- http://www.omgsysml.org/The_SysML_Modelling_Language.pdf
- www.incose.org/orlando/Attach/200704/**Why_SysML.ppt**
- http://www.sysml.org/partners.htm
- http://www.mel.nist.gov/msidlibrary/doc/sysmlactivity.pdf
- http://www.iasted.org/conferences/2007/innsbruck/se/pdfs/UML2point0_Tutorial_brief_.pdf
- ihase.wikispaces.com/space/showimage/**SysML**+**Introduction**3.ppt
- http://www.elsevierdirect.com/v2/companion.jsp?ISBN=9780123786074