SysEng 6542
Model Based Systems Engineering

Systems Modeling Language (SysML)

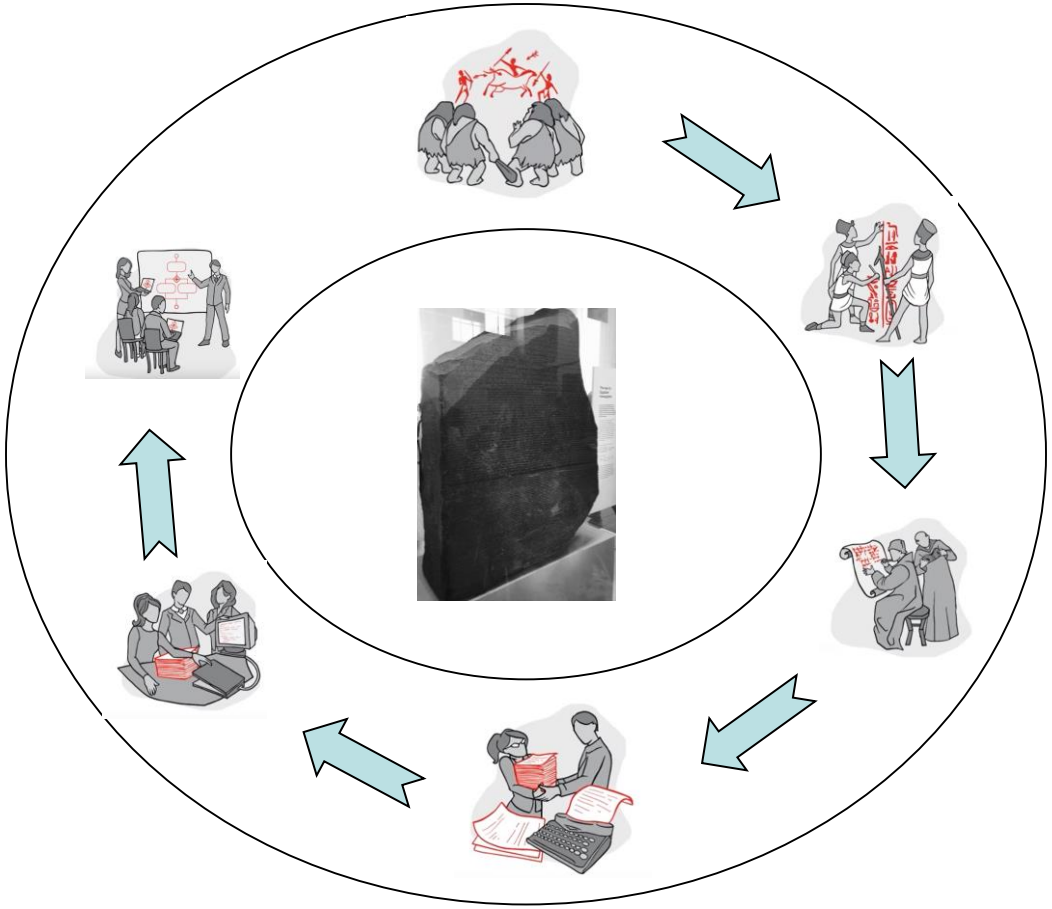
Dr Quoc Do

MBSE Methodology

- A Systems Engineering Language - SysML
- A set of SE Processes
- A Systems Engineering Tool(s)

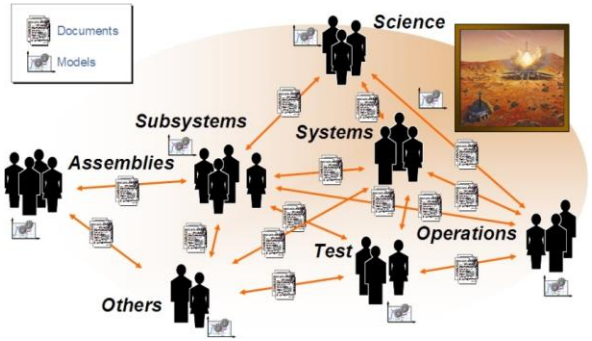
Systems Modeling Language

- Graphical representation of system information
- Diagrams used to contain information

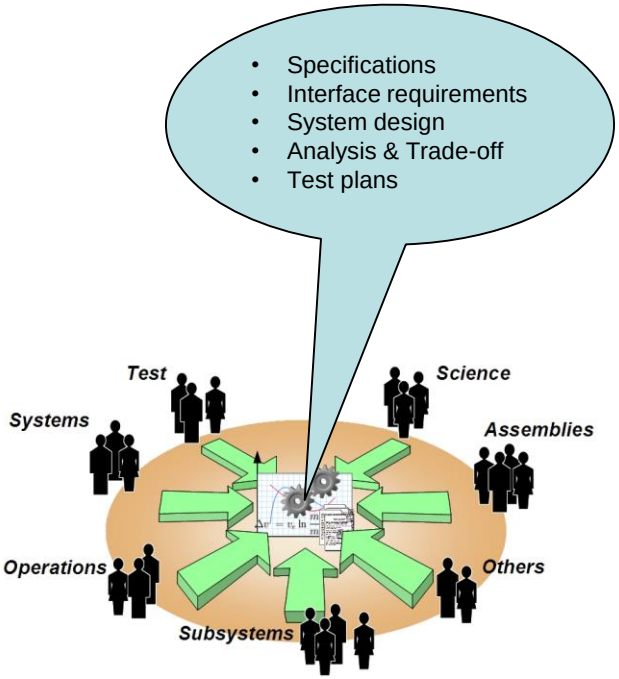


Systems Modeling Language

- Systems modelling language enables model-centric Systems Engineering practice

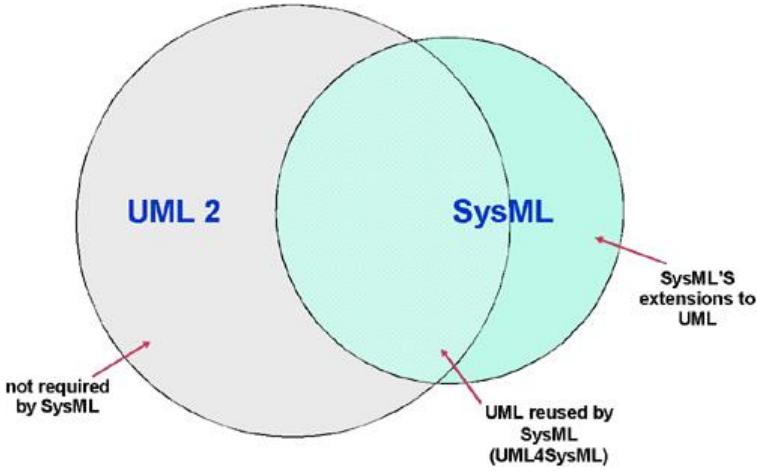


Transition



SysML and UML

- SysML is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities.
- SysML is an extension of UML and has four key pillars:
  - Requirements
  - Structure
  - Behavior
  - Parametric

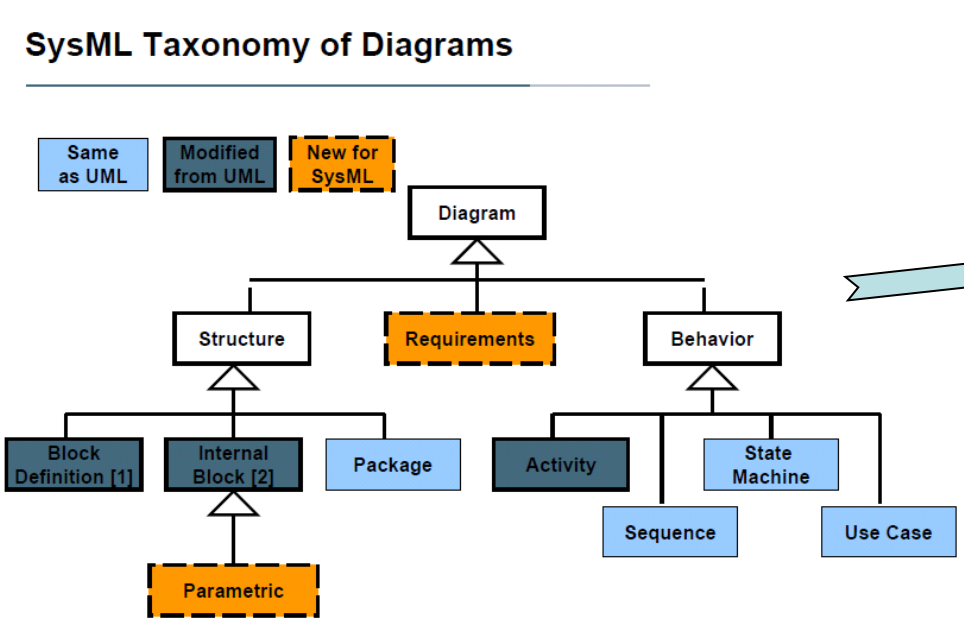


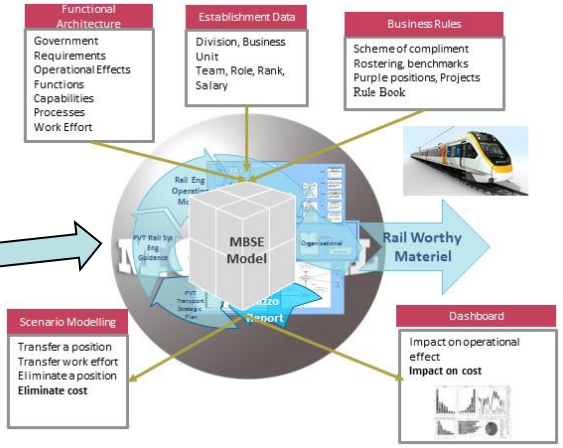
The SysML Architecture

- This architecture provides a means to transfer systems engineering domain information into a metamodel
- This allows for the representation of systems in a uniform model representation
- Enable reuse of models
- Adoption of standards ensures interoperability (…)
- Complexity management

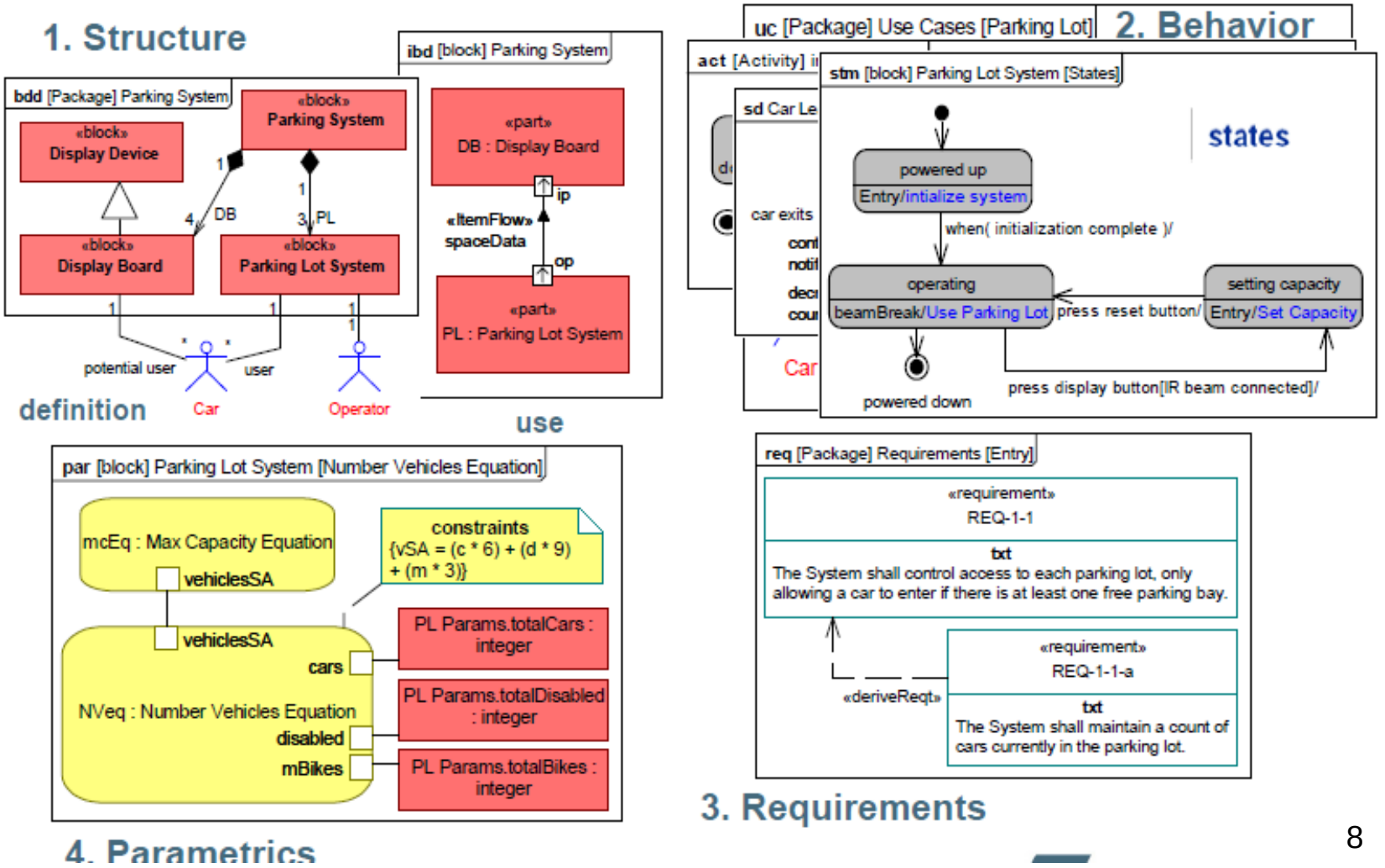
Diagrams

- Graphical representation of the system information
- 9 types used in SysML
- Classified by the type of data they contain

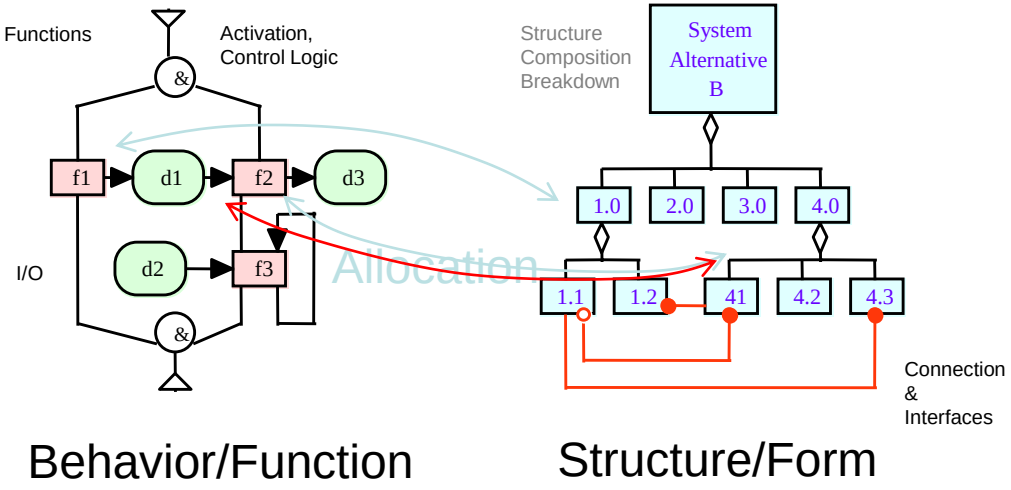




The Four Pillars of SysML



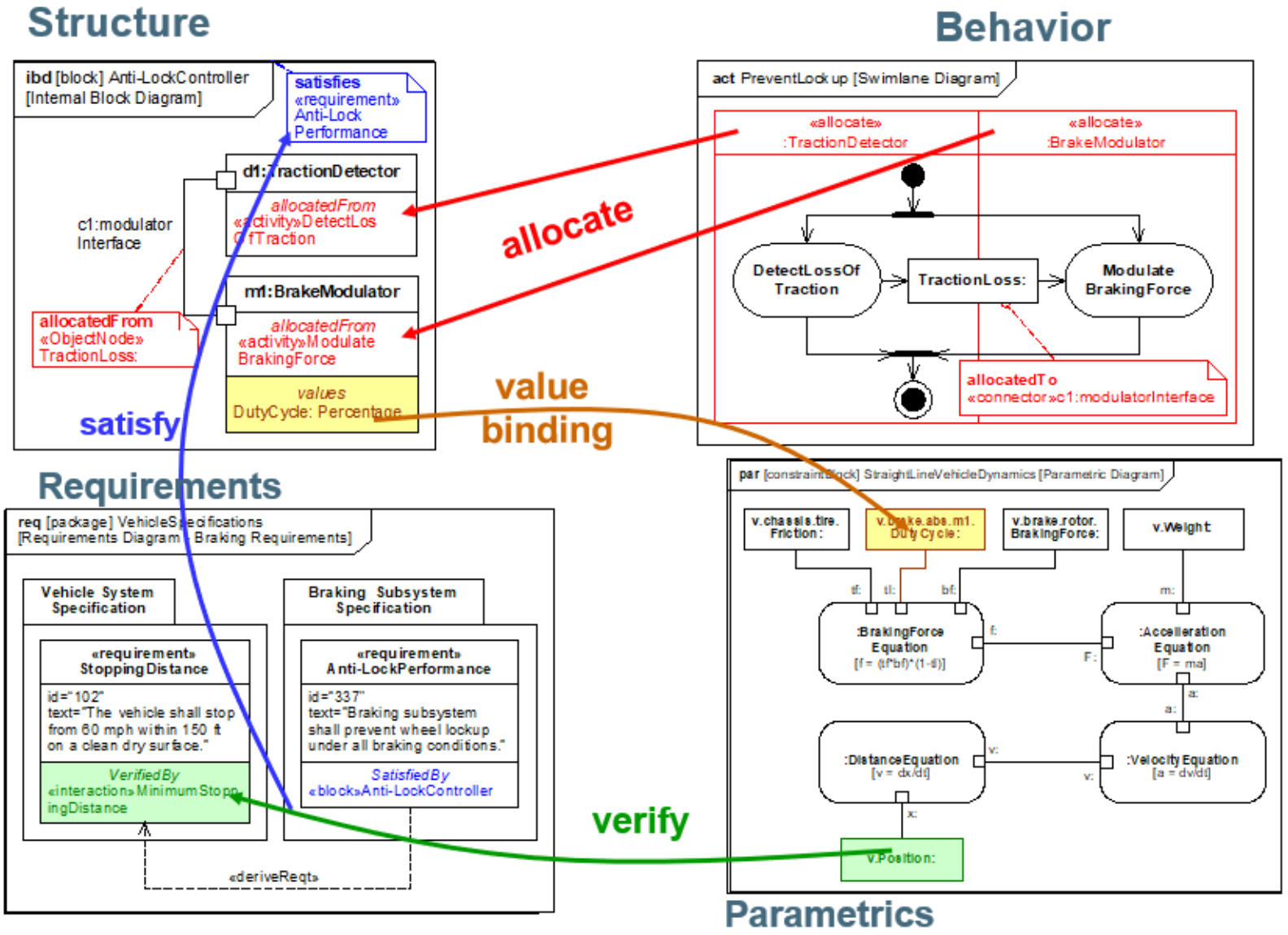
Allocation – Function to Physical Elements



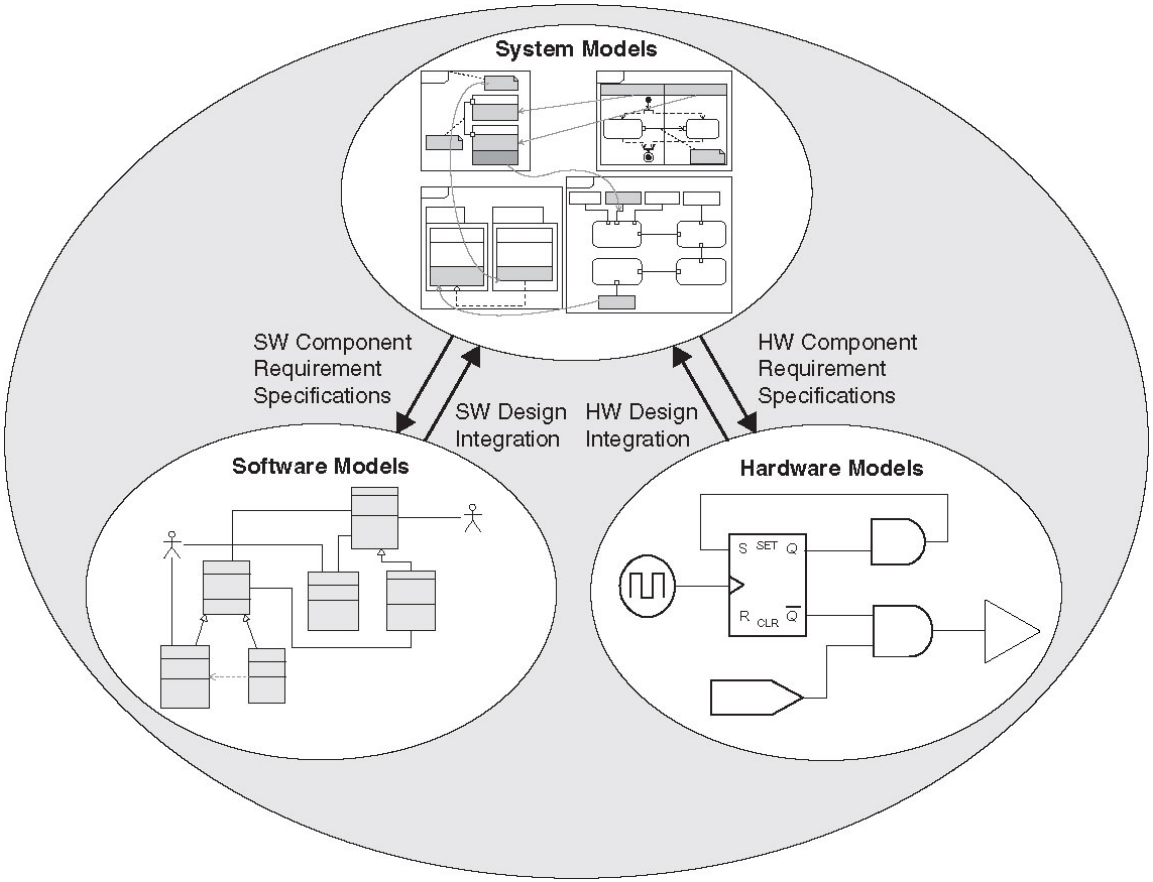
"functions" explicitly allocated to components

"I/O" explicitly allocated to interfaces

SysML: Cross Connecting Model Elements



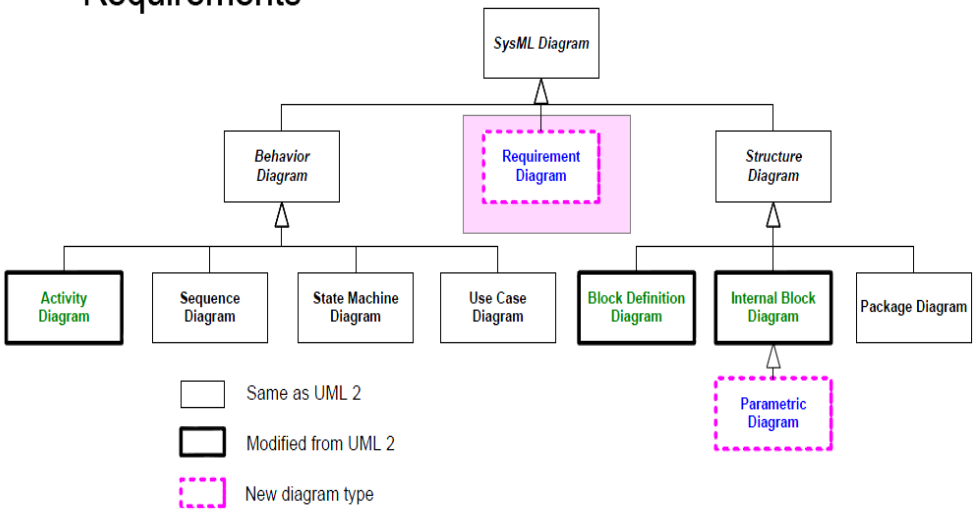
Connecting Models Cross Disciplines



Overview of SysML Diagrams

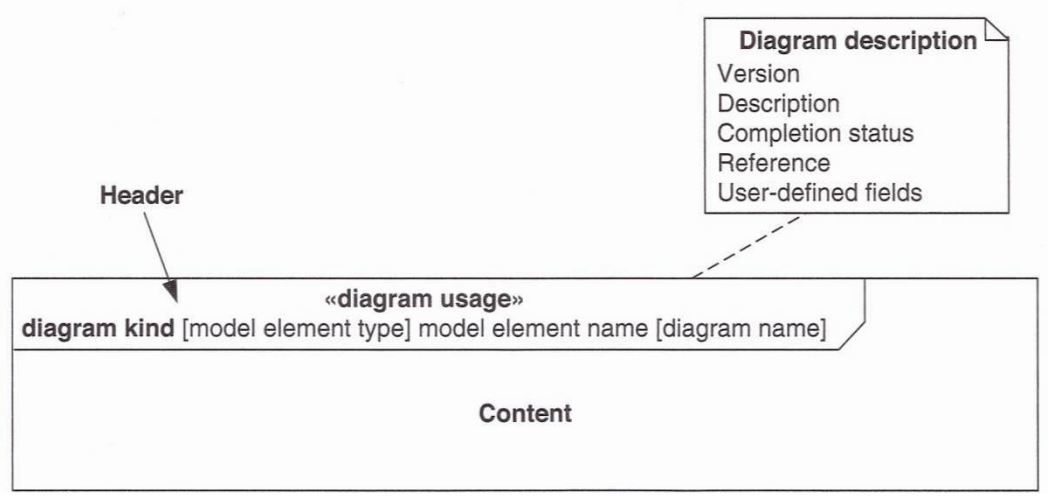
- Anatomy of a Diagram
- Diagram Types:
  - Requirement (req)
  - Structure
    - Block Definition (bdd)
    - Internal Block (ibd)
    - Parametric (par)
    - Package (pkg)
  - Behavior
    - Activity (act)
    - Sequence (sd)
    - State Machine (stm)
    - Use Case (uc)

- Allocations
- Requirements



Anatomy of a Diagram

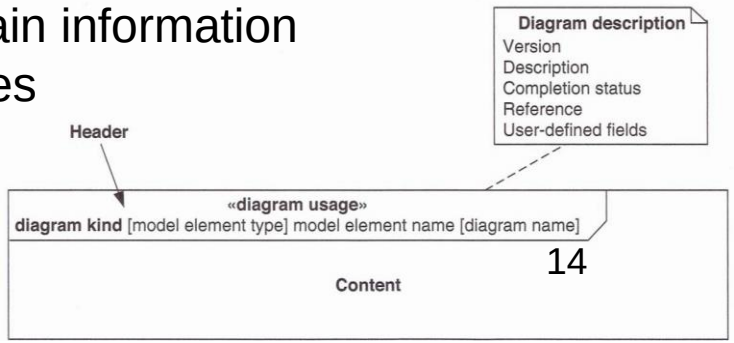
- Frame
- Header
  - Name
  - Usage
- Description
- Content



***A Practical guide to SysML*, Friedenthal et al.**

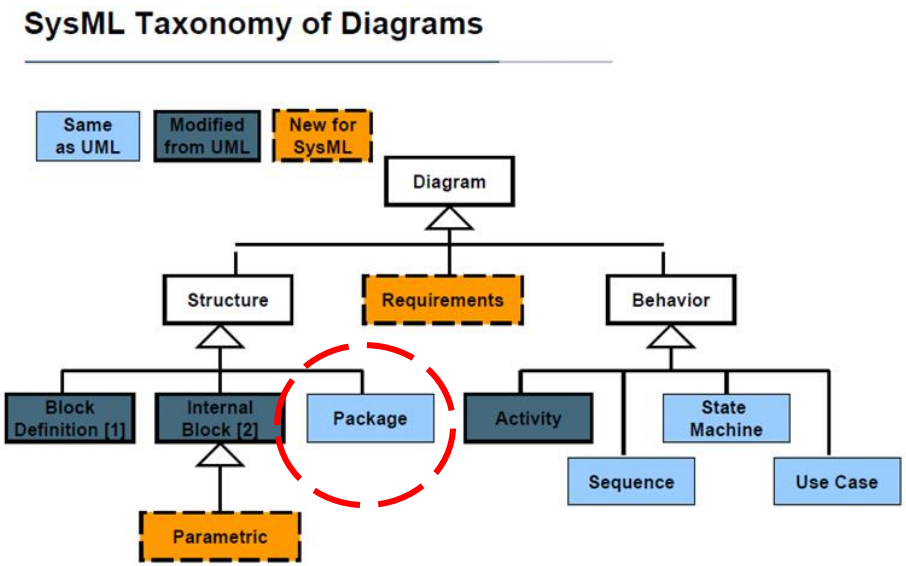
Anatomy of a Diagram

- Frame – Visible context of the diagram content
- Header/Name – Includes the information to identify the diagram
- Header provides:
  - Diagram kind, Model element type, Model element name, Diagram name, Diagram usage
  - Diagram kind – diagram classification
  - Model element type
    - Model element name
    - Diagram name
    - Diagram usage
- Description – Optional note that provides 'bookkeeping' information (version, status, etc.)
- Content – The actual information on the elements of the model.
- The content area contains the graphical elements representing model elements: Node: Contain information (symbols, icons, etc.) describing system entities



SysEng 6542
Model Based Systems Engineering

Modelling Physical Systems



Organizing the Model

- Model elements are placed in containers
- Containers can be nested
- Containers can be thought of as component entities.
- Ease of access and navigation between elements
- Reuse of elements
- Assists configuration management
- Reuse – Libraries
- Views and Viewports
- Import relationships

Model Libraries

- Package hierarchy constructed with the intent for it to be reused.

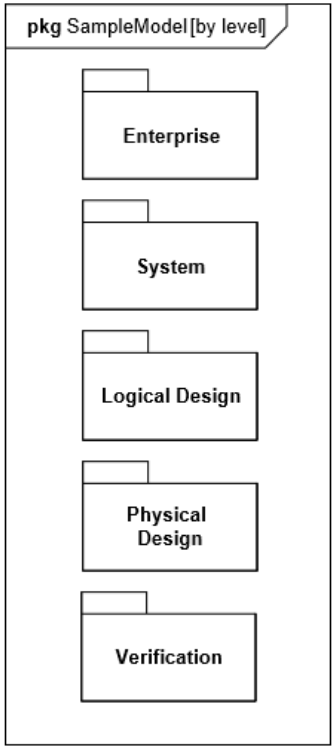
- Libraries can be used for
  - COTS items
  - Standard definitions
  - Company specific products
  - Etc.

Package Hierarchy

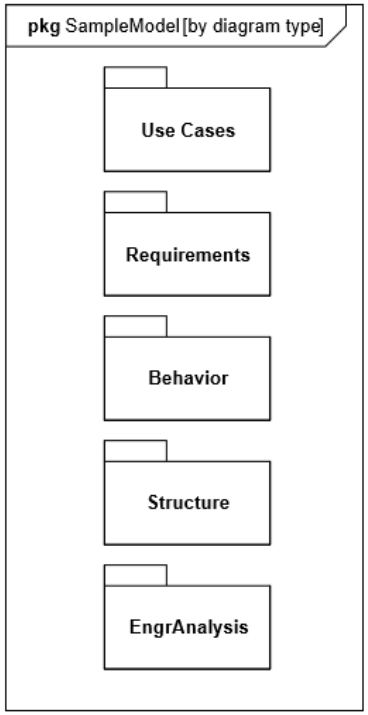
- Gives a standard organization of model data
- Allows data to be arranged in domain specific methods
- Critical to access control, configuration management, data exchange, etc.

Package Diagrams

- Provide a means to organize system model data
- The SysML model is a top-level package

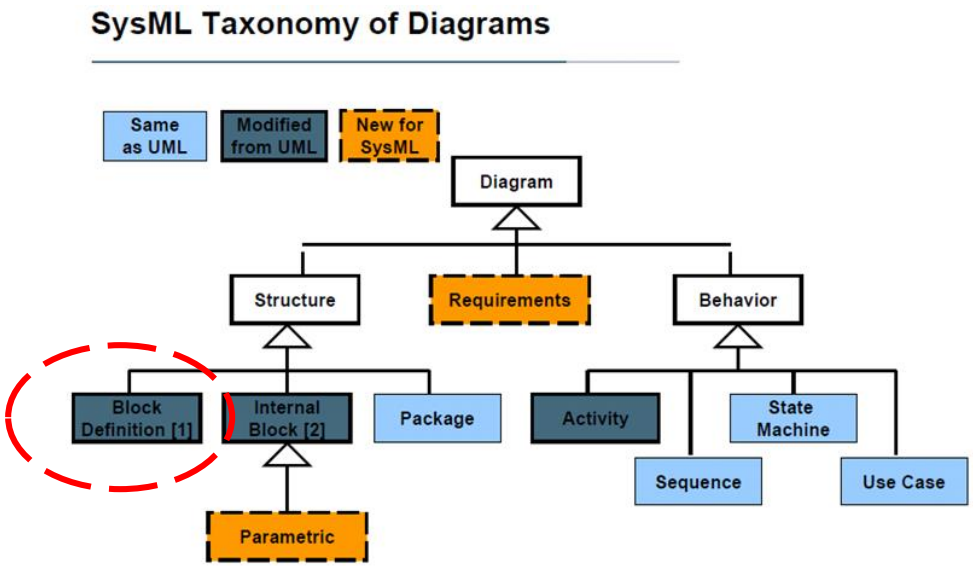


OR



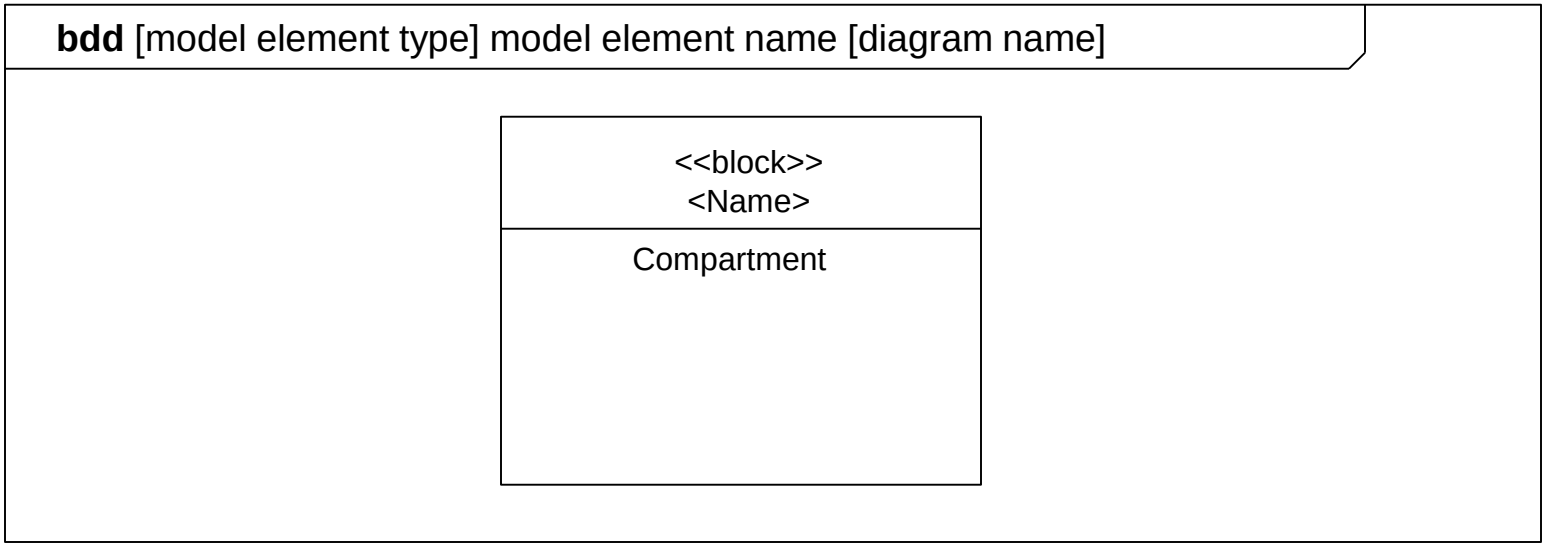
SysEng 6542
Model Based Systems Engineering

Modelling Physical Systems

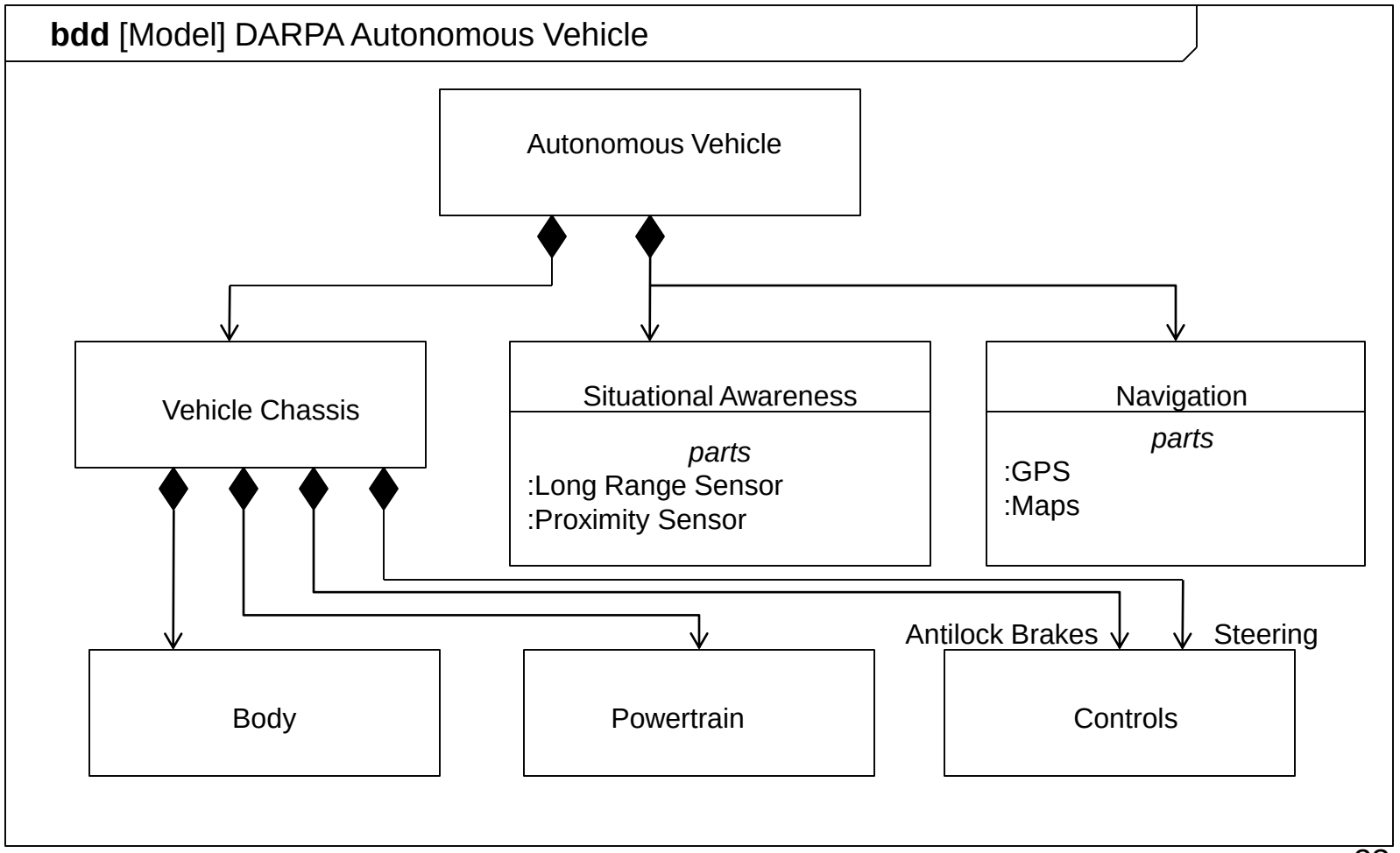


Block Definition Diagram

- Modified UML Structure (Class) Diagram
- Used to define blocks and their structural relationships with other blocks (HW, SW, system, facility, etc.…)
- Describes the pieces used to construct the model
- Describe a set of similar instances
  - "Engineering objects"

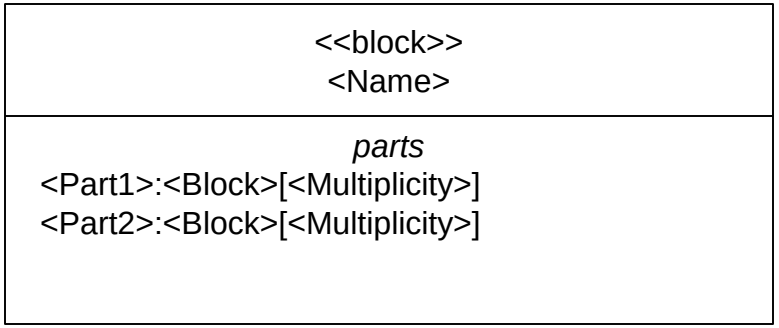


Blocks showing hierarchy



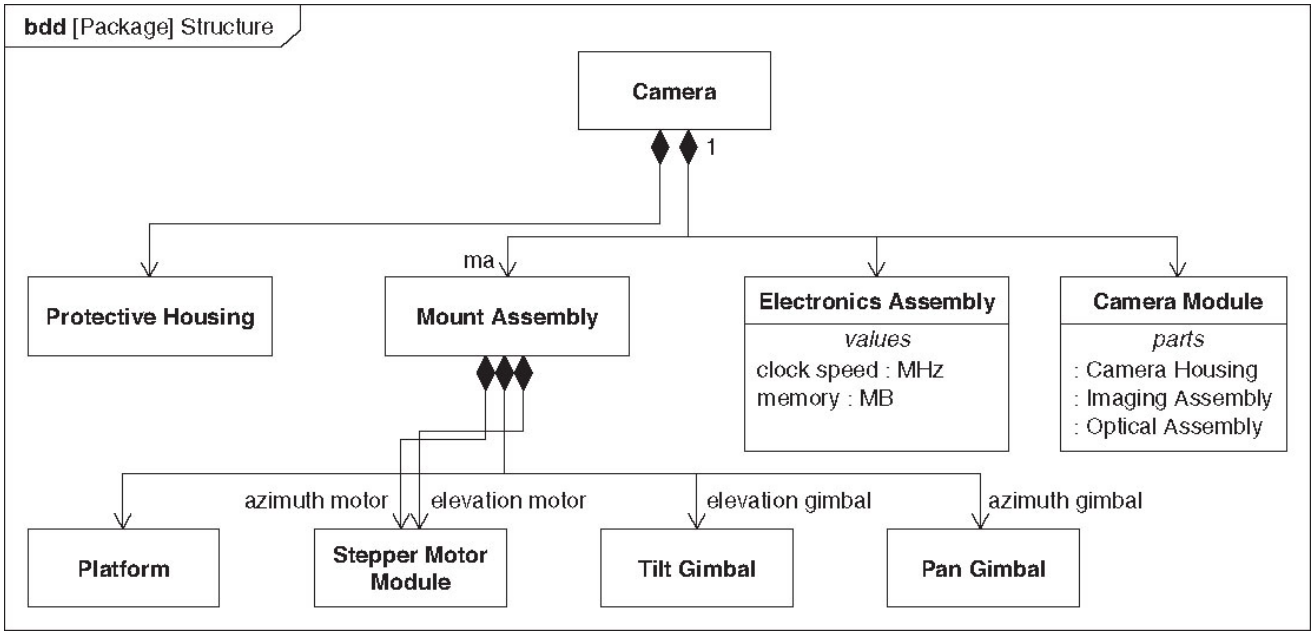
Blocks

- Compartments: hold additional information about the node
  - Part properties
  - Reference properties
  - Value properties
  - Operation
  - Reception
- Part properties (*part*)
  - Describe composition relationships between blocks
  - Contains part name, block, and number of instances (multiplicity)



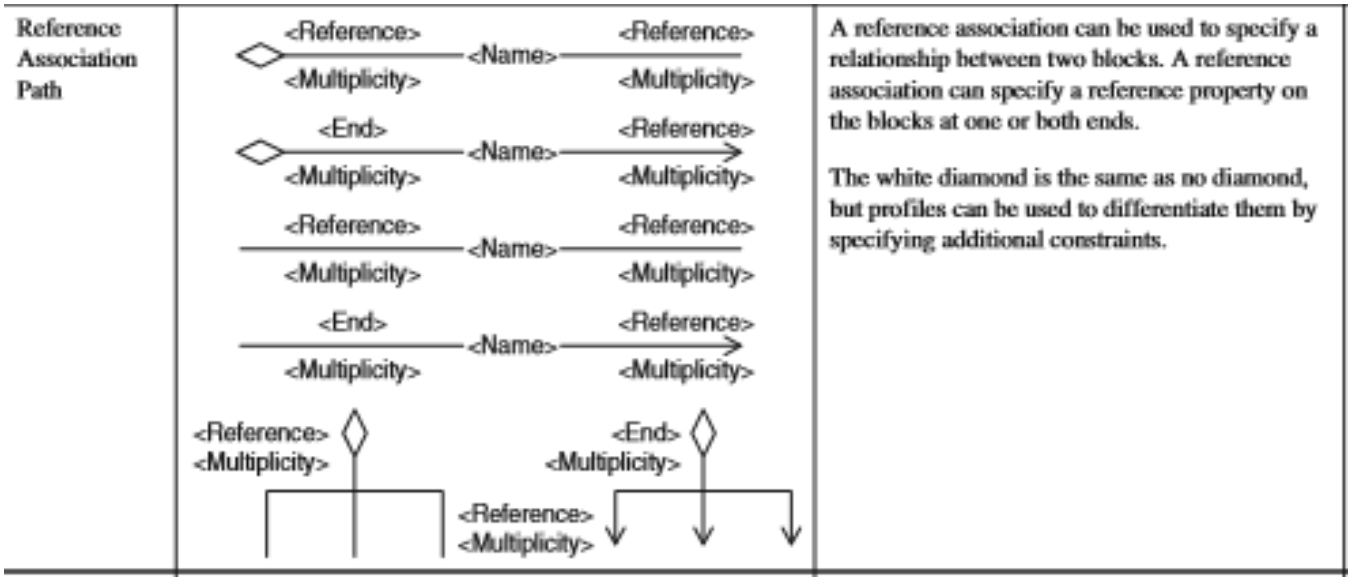
Composite Associations

- Describes whether the part can exist without the block it is associated with
- Multiplicity at whole end is either 1 (needed by part) or 0..1 (not needed)

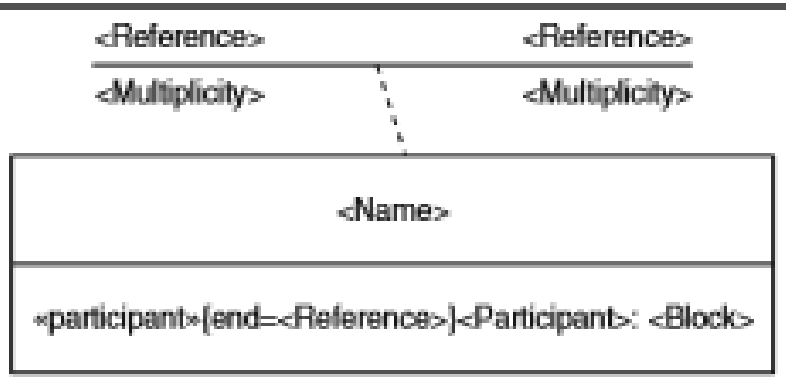


Reference Properties

- Reference properties (*reference*) describe additional relationships
- They describe additional associations between blocks
- Can be shown in a separate 'hierarchy'



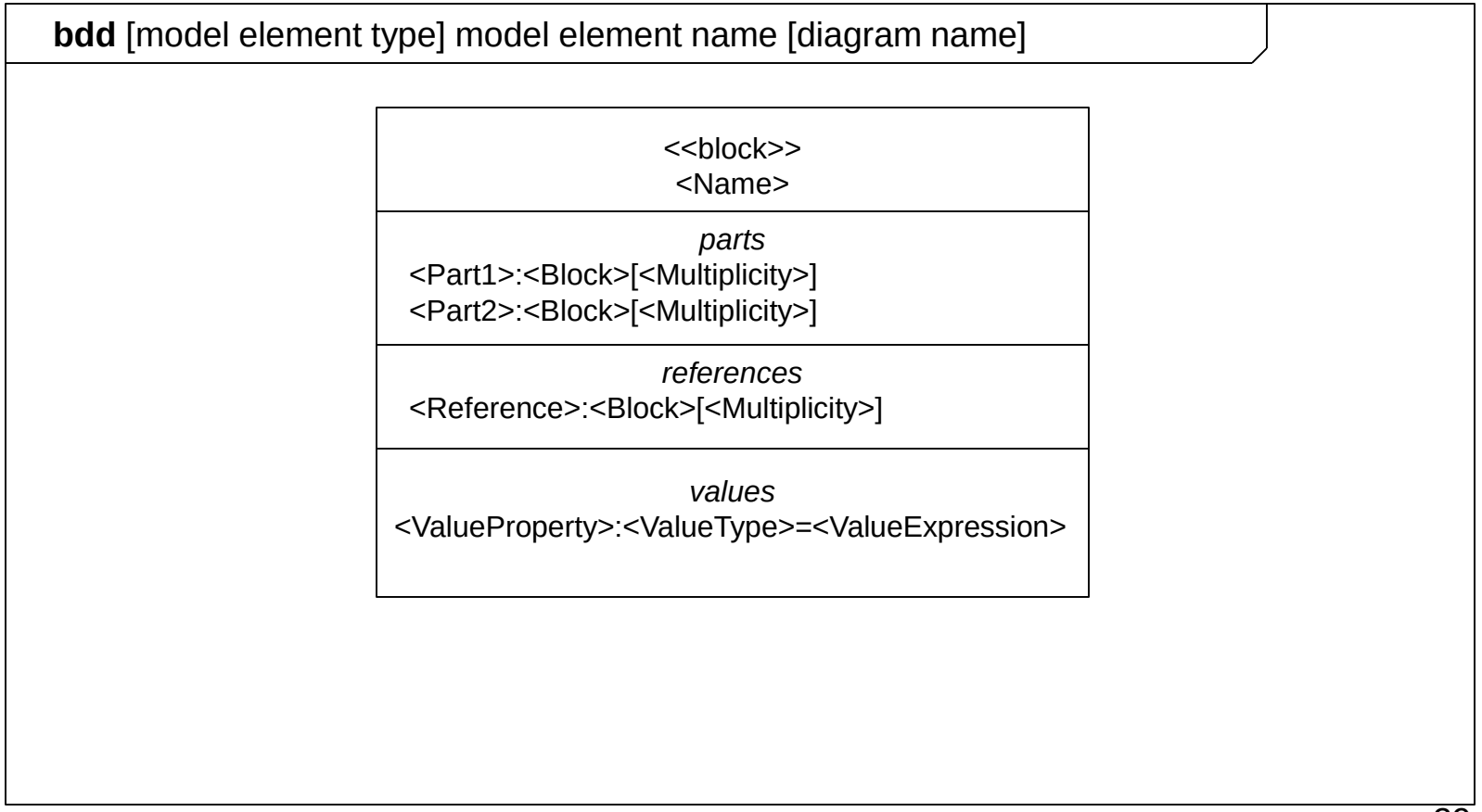
Block Associations

Association Block Path and Node		An association block, as the name implies, is a combination of an association and a block, so it can relate two blocks together but can also have internal structure and other features of its own. Participants are placeholders that represent the blocks at each end of the association block, and are used when it is desired to decompose a connector.
Generalization Path		A generalization describes the relationship between the general classifier and specialized classifier. A set of generalizations may either be {disjoint} or {overlapping}. They may also be {complete} or {incomplete}.

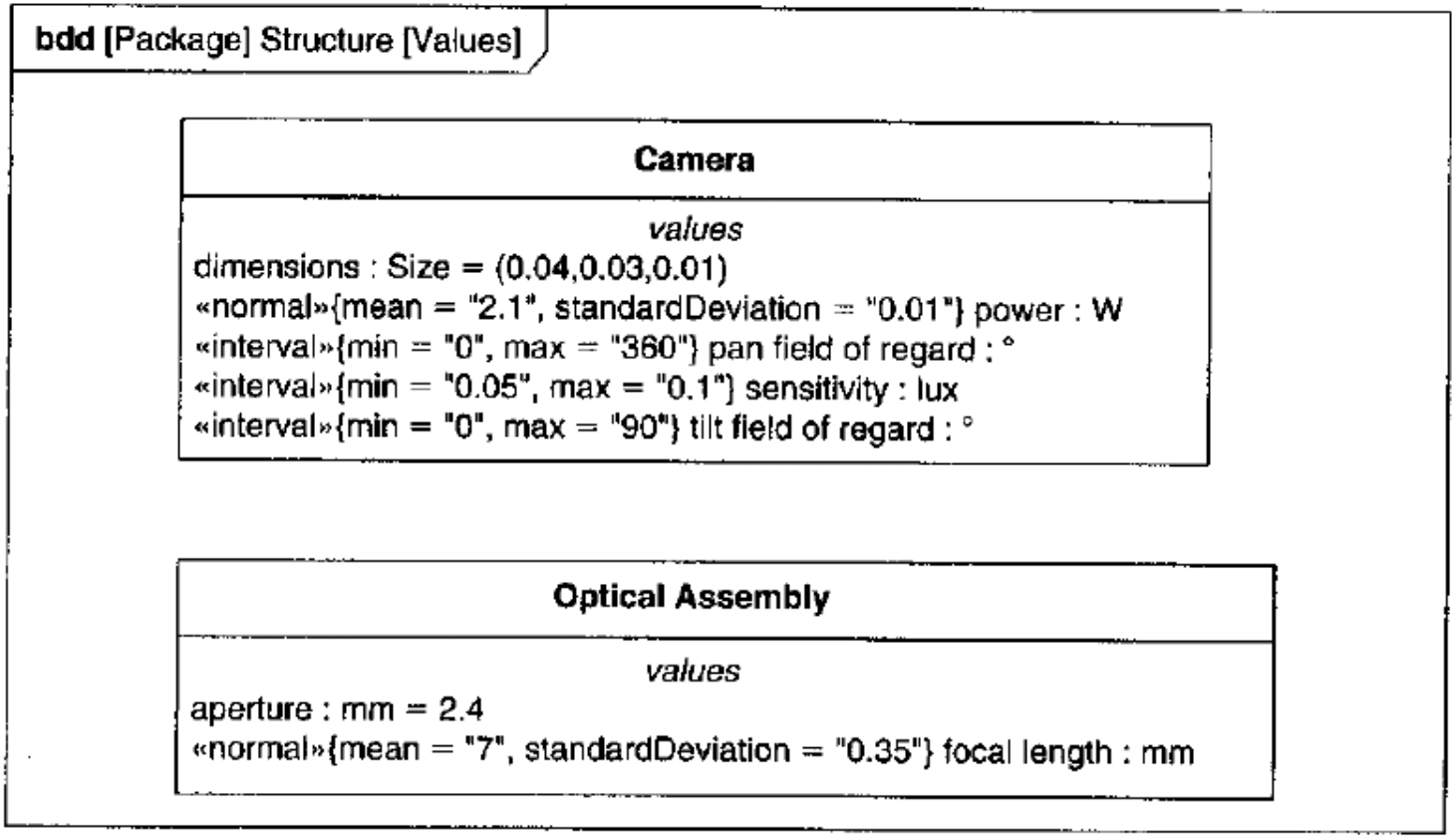
Value Properties

- Value properties (*value*) add quantitative properties to blocks
  - Value types describe the values for the quantities (think of it as units)
  - Can take on primitive type (integer, string, etc.), enumeration, or custom representations
  - Can read in model libraries to apply measurement systems (English, SI, etc.)
  - May also be used for derived properties
    - Value properties with values that are calculated

Blocks


bdd [model element type] model element name [diagram name]
<<block>>
<Name>
parts
<Part1>:<Block>[<Multiplicity>]
<Part2>:<Block>[<Multiplicity>]
references
<Reference>:<Block>[<Multiplicity>]
values
<ValueProperty>:<ValueType>=<ValueExpression>

Block - Value Properties



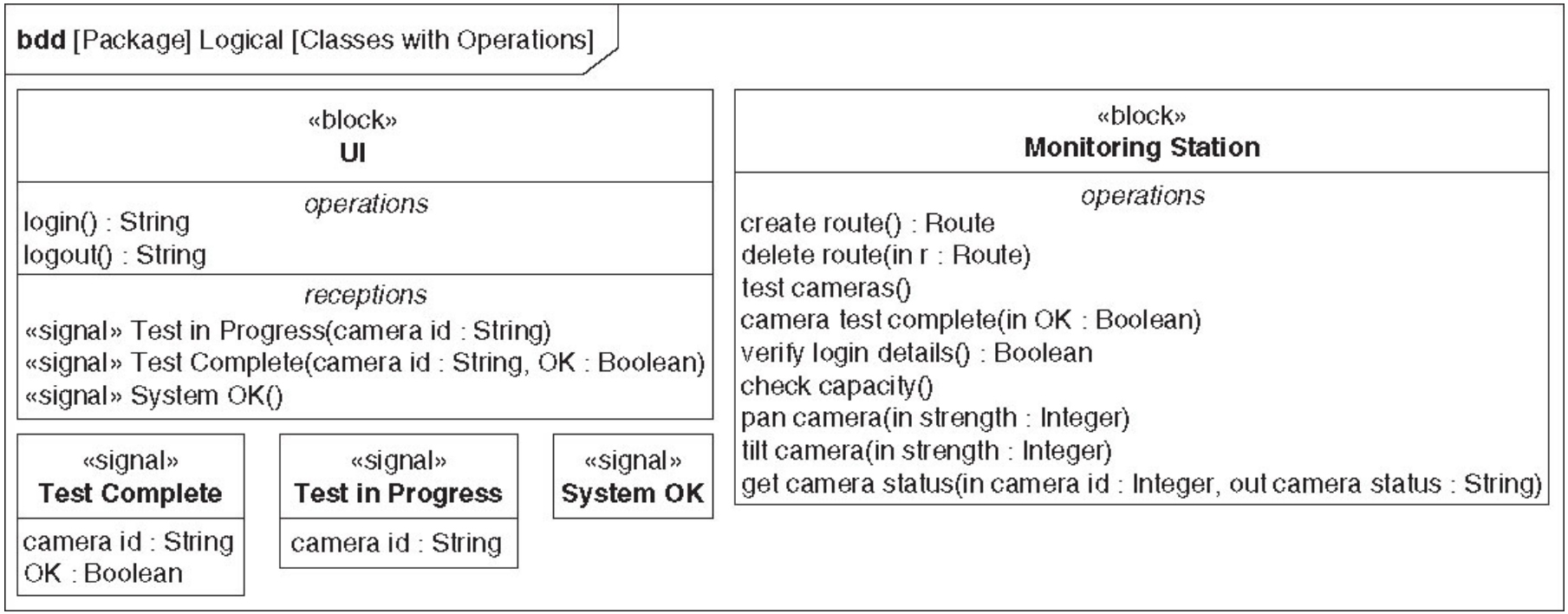
Modeling Block Behavior

- Main behavior of block can be shown in behavioral features
  - Operations
  - Receptions
- Actual behavior shown in function, state machine, activities etc.…

- Receptions are asynchronous requests
  - Each reception is associated with a signal

- Operations are synchronous requests
  - Parameters are defined and passed with request

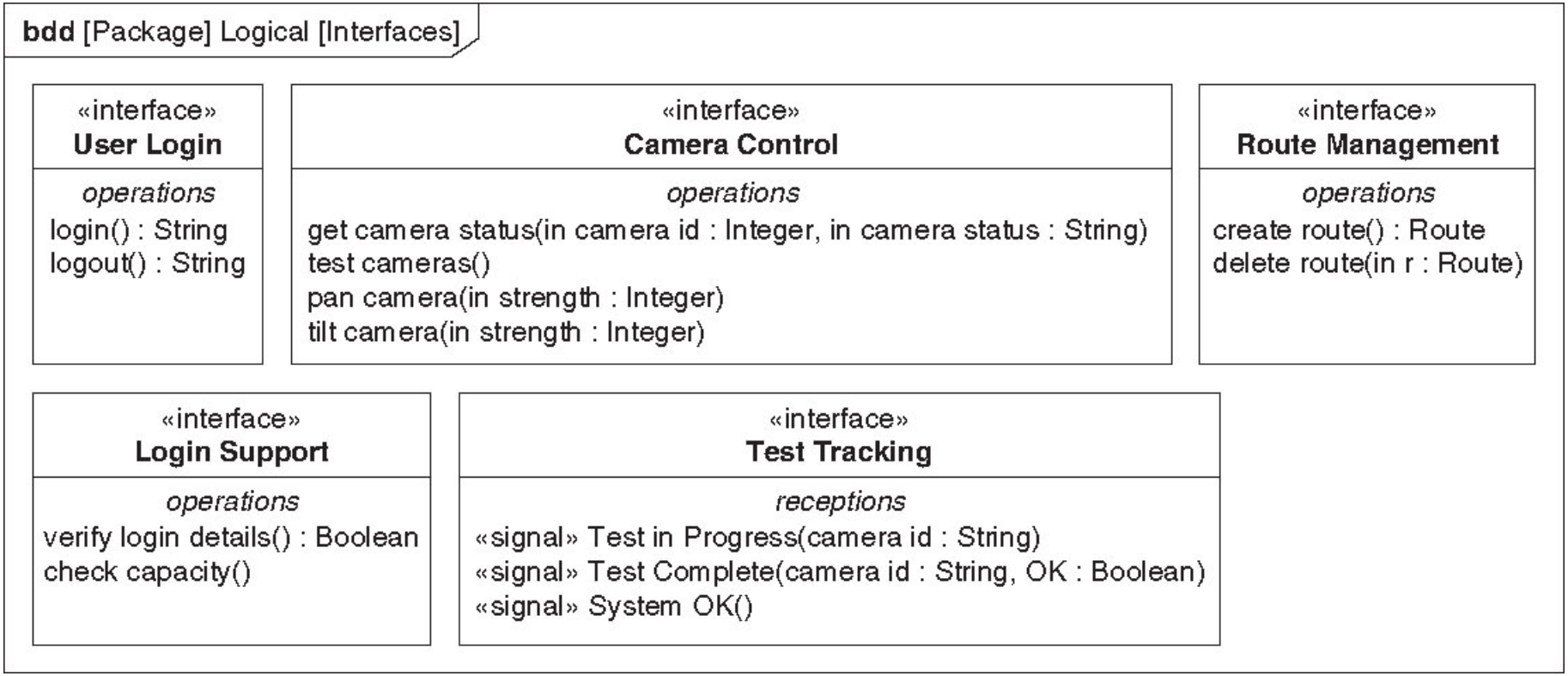
Modeling Block Behavior



Modeling Block Behavior

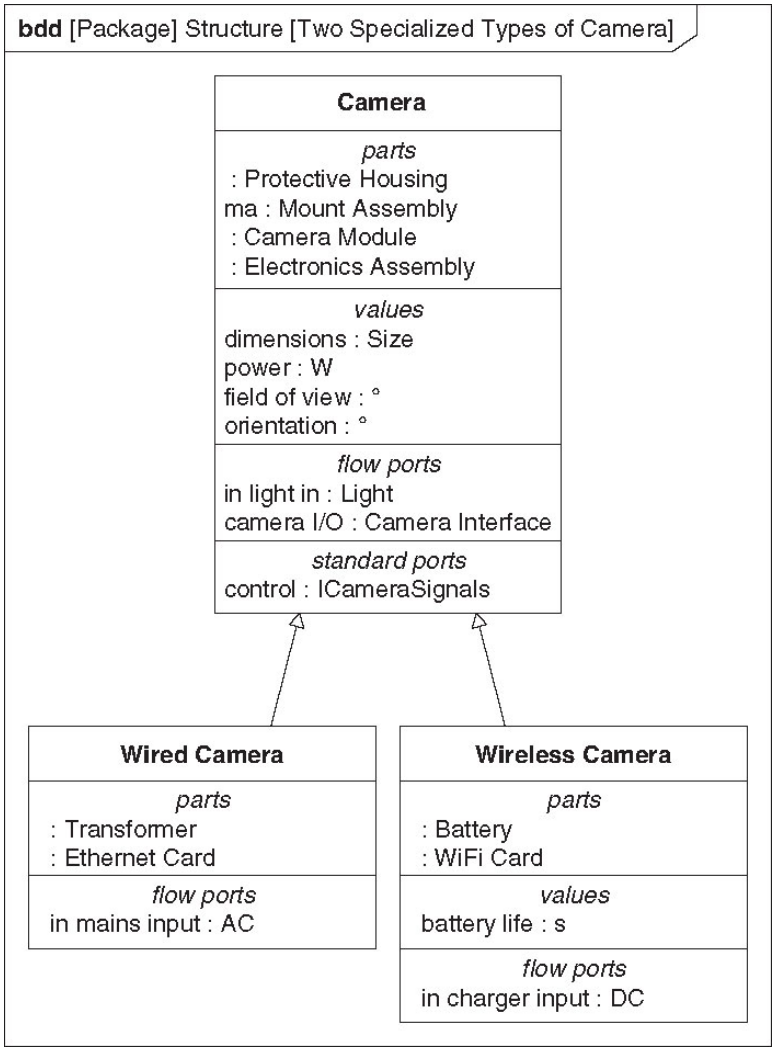
- Interface definitions for ports can also define behavior
  - Required interface: operations required by block to realize behavior
  - Provided interface: operation that a block provides

Modeling Block Behavior



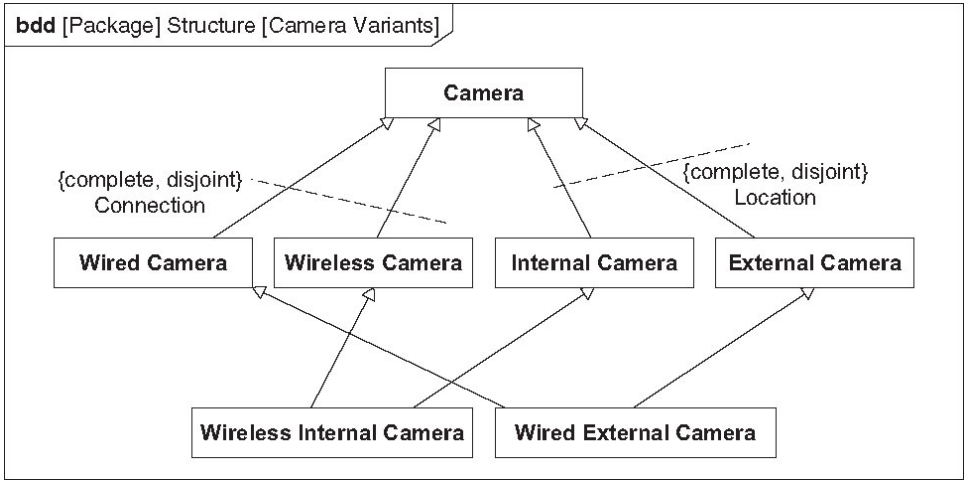
Modeling Classification Hierarchies

- Block definition diagrams can be used for classification
  - Superclass : general classifier
  - Subclass : specialized classifier
  - Generalization : relationship between superclass and subclass



Modeling Classification Hierarchies

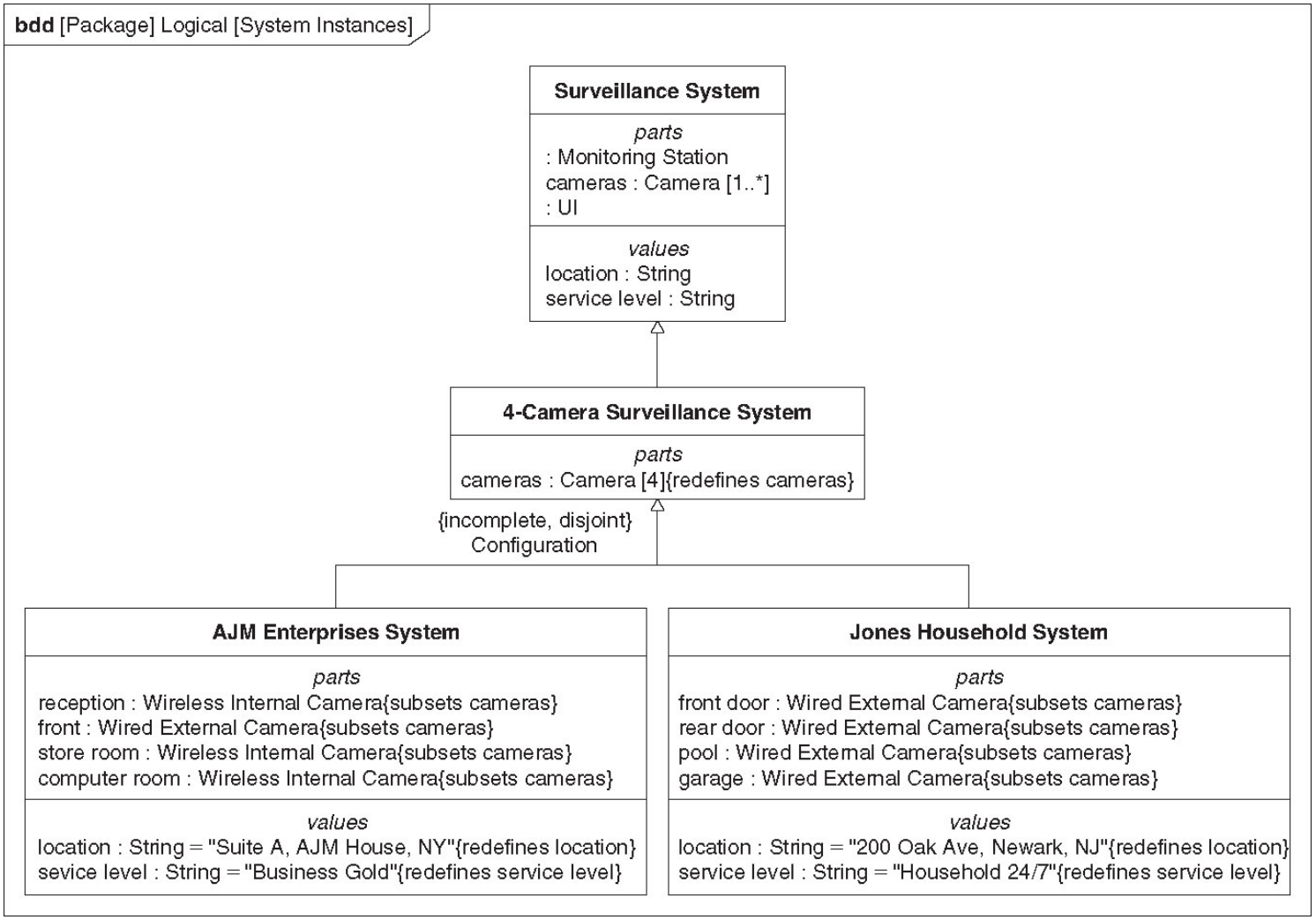
- Using subclasses allow for:
  - Supplying additional detail
  - Redefinition of features
    - Change multiplicity
    - Add or change initial value
    - Add or change probability distribution
    - Change property type to a more restrictive property



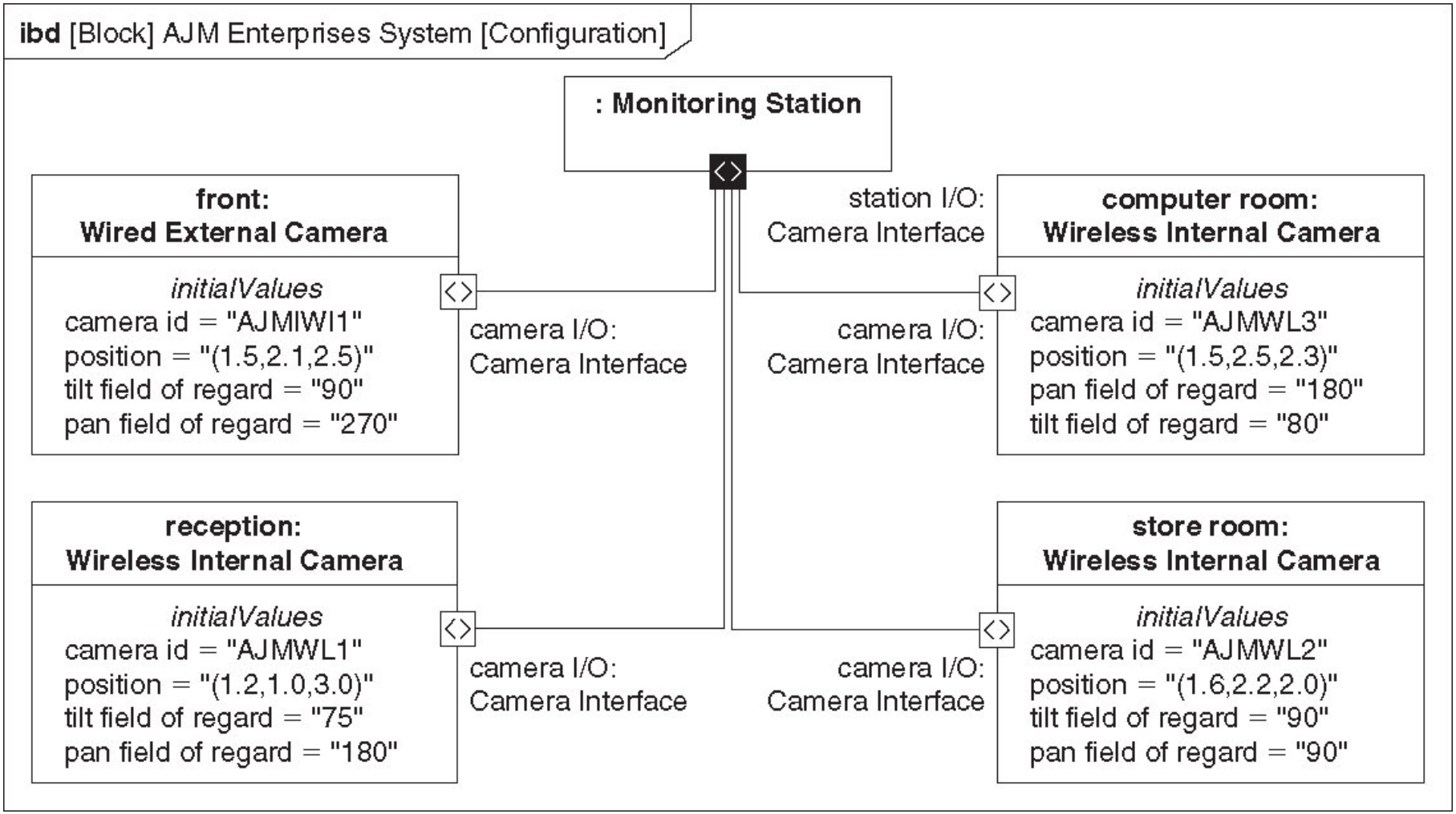
Generalization

- Similar to class membership
- Provides description of relationship
  - Coverage : complete or incomplete
  - Overlap : can instance be a member of more than one subclass

Modeling Configurations

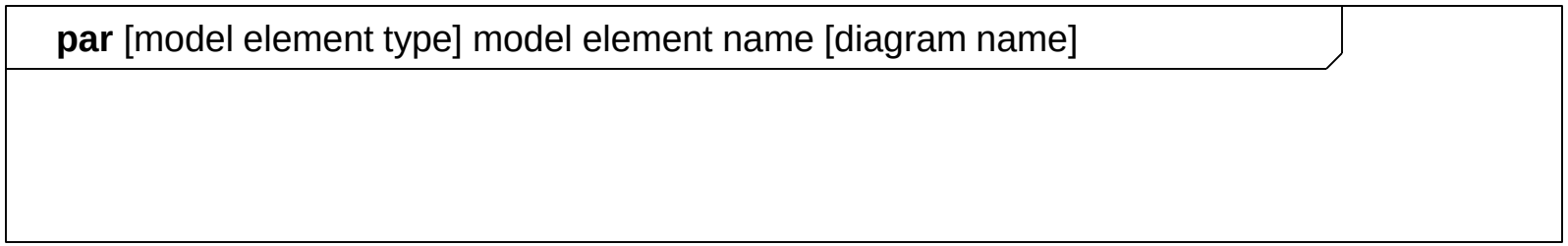


Modeling Configurations

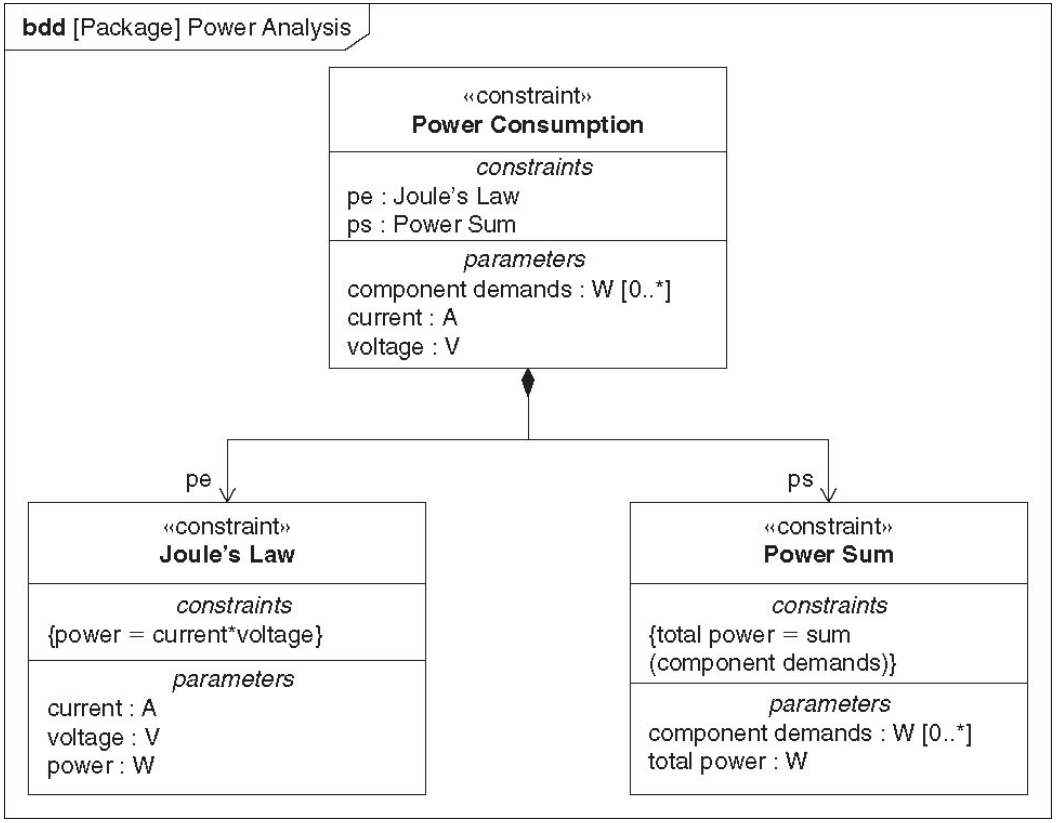


Parametric Diagram

- New Structure Diagram
- Contains information to create a system of equations to constrain the block properties

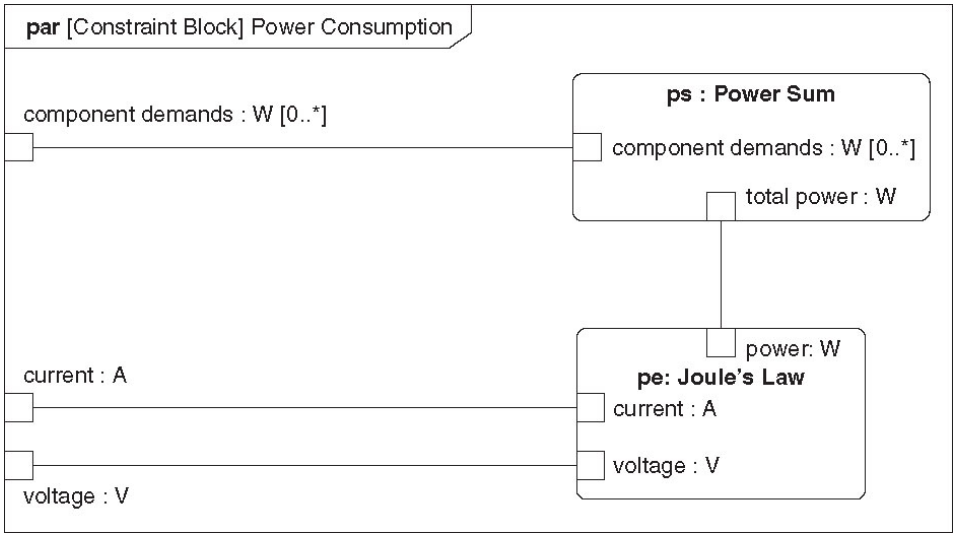


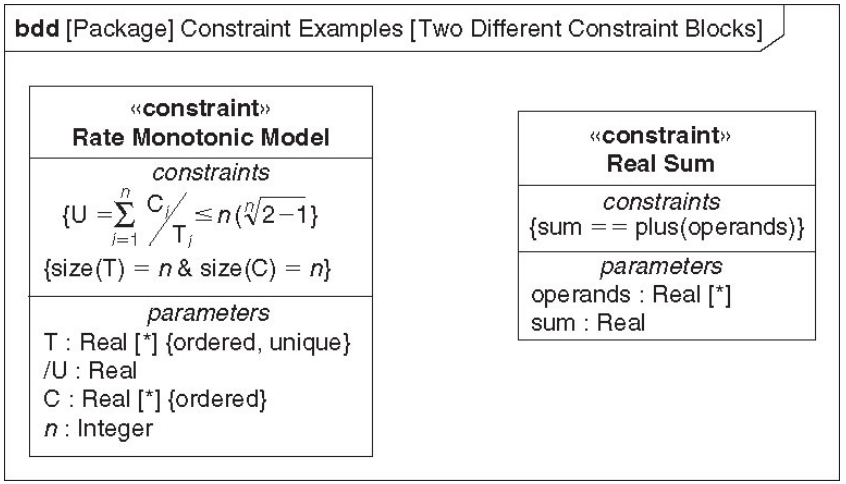
Parametric Diagrams



Parametric Diagram

- Can describe interfaces with external software
  - Can be given in constraints compartment or as an attached note
- Constraints can also be given in a separate block

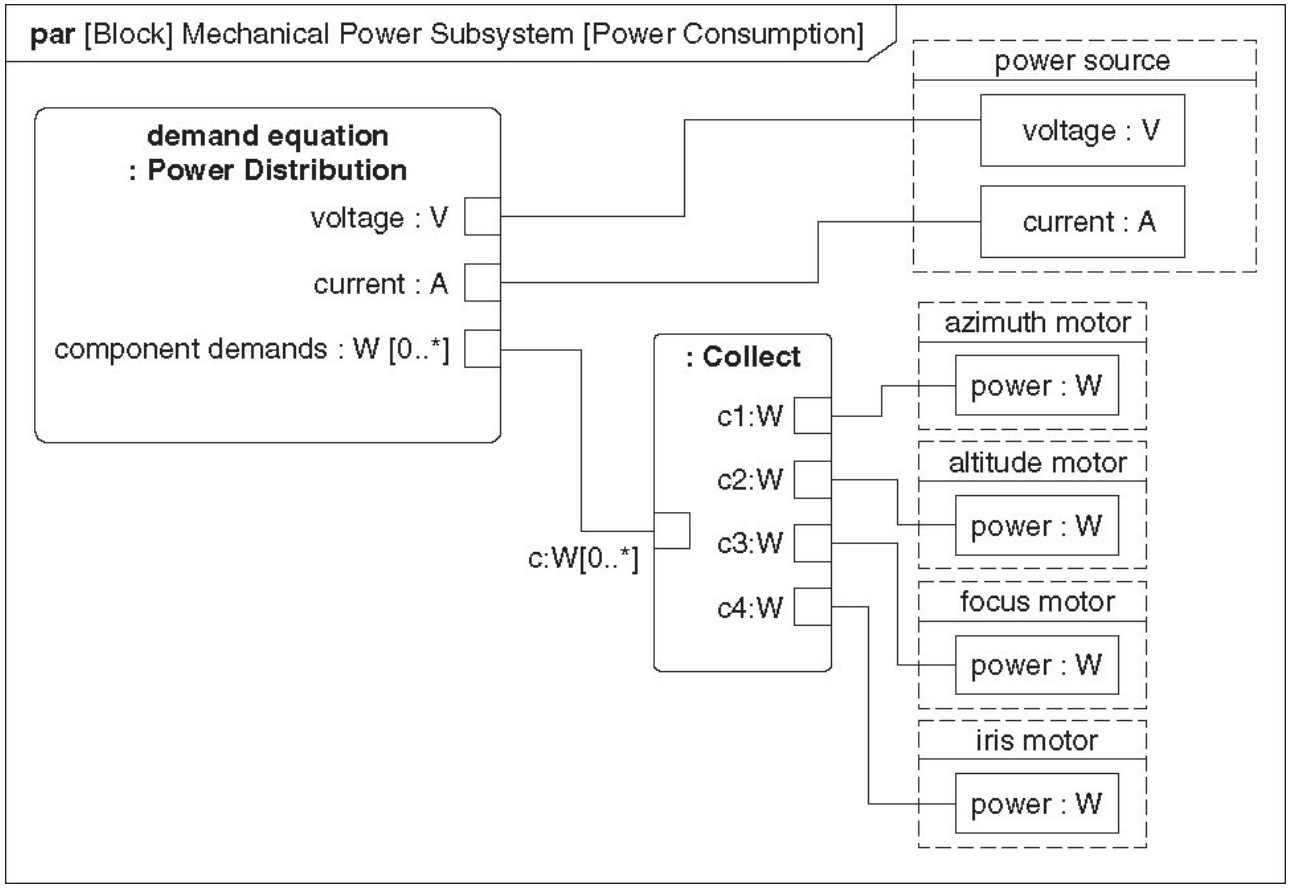




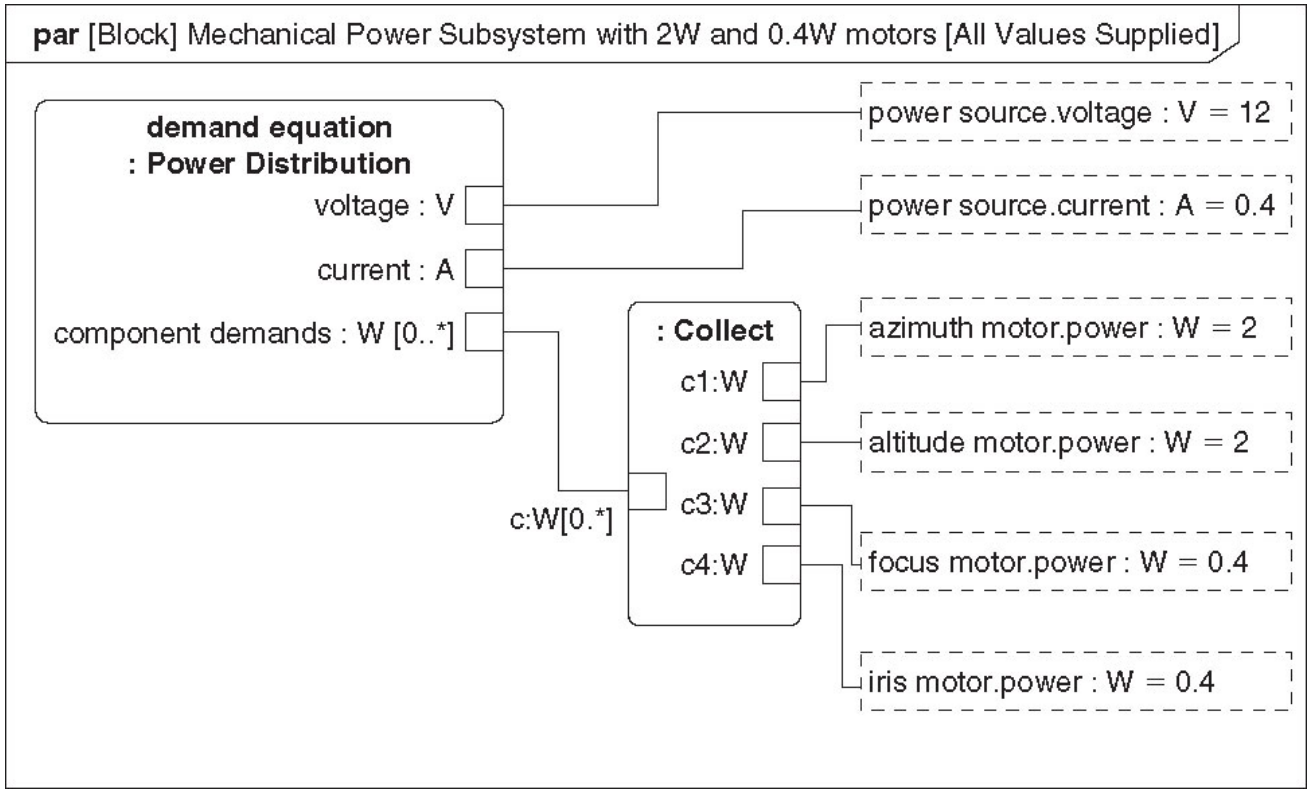
A note on external solvers

If an external solver is to be used, initial values will typically need to be specified

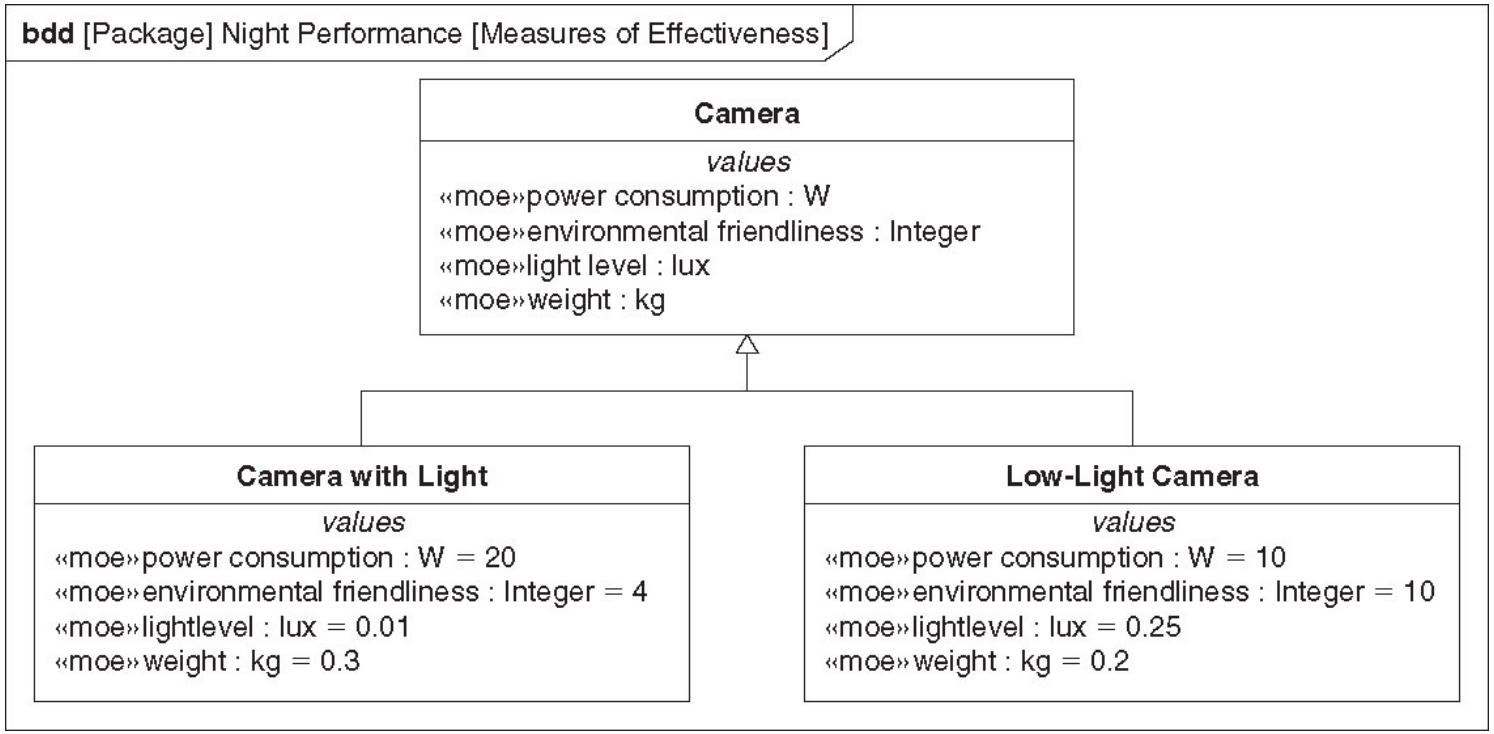
Parametric Diagram …



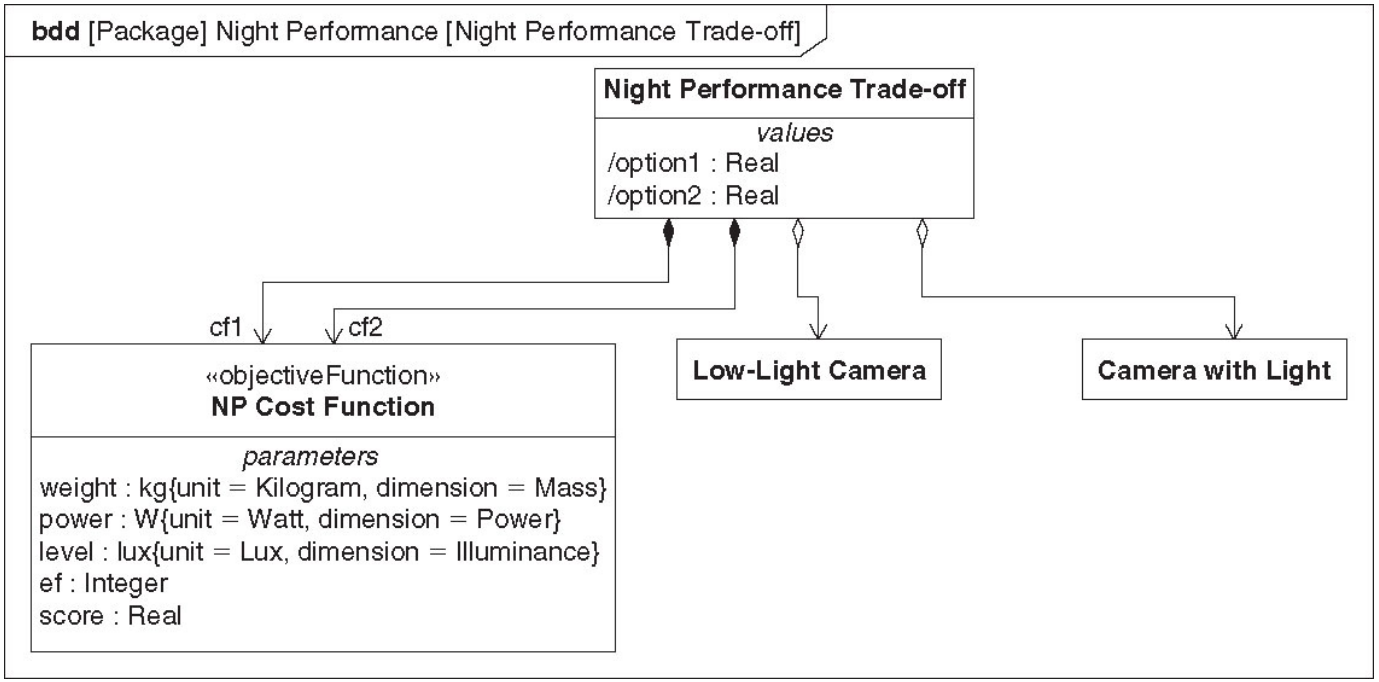
… with initial values



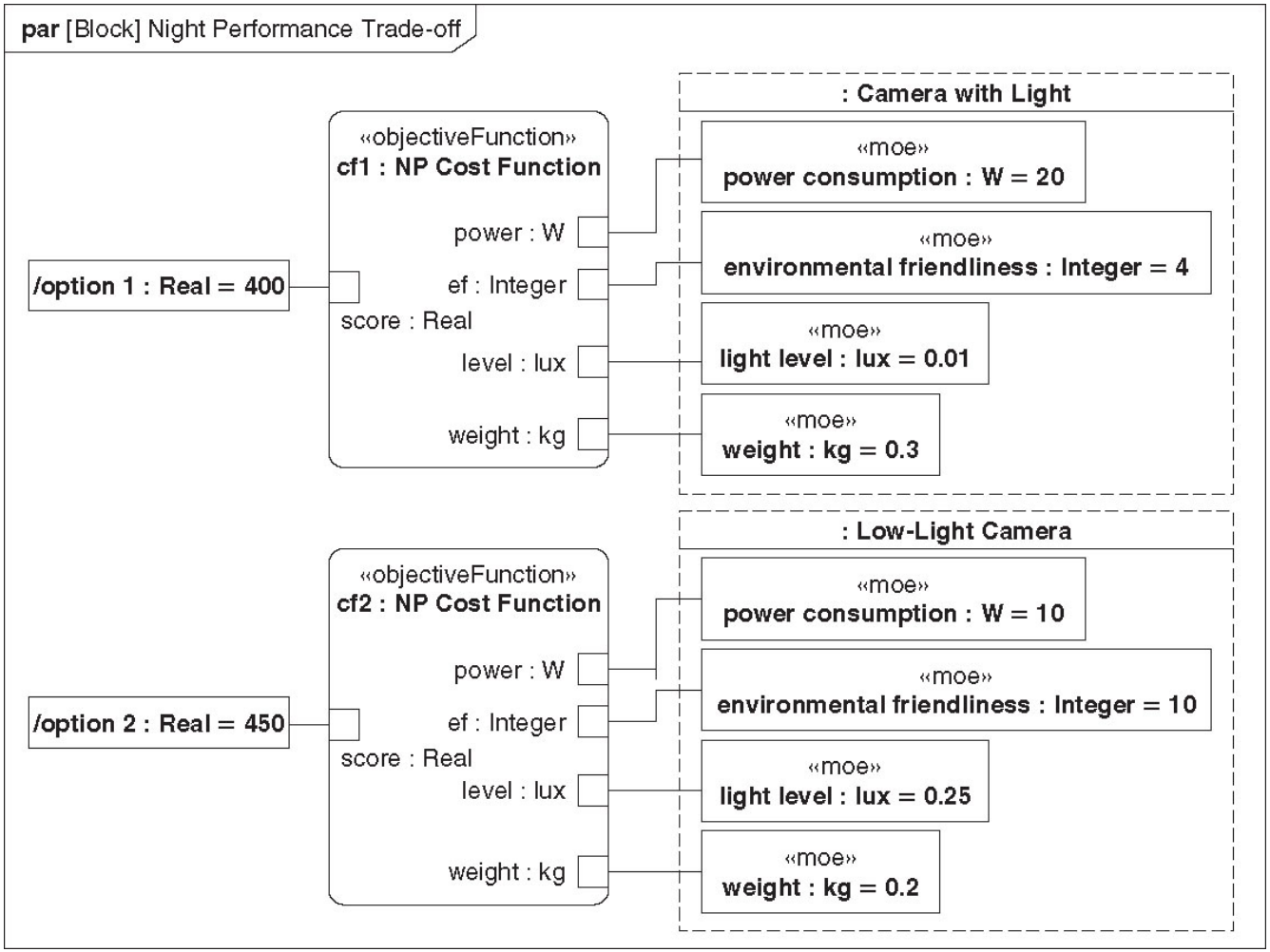
Parametric Diagrams for Trade Studies



Parametric Diagrams for Trade Studies



Parametric Diagrams for Trade Studies



Next Time

- More on SysML Behaviors
- Assembling models

Program Completed

Missouri University of Science & Technology