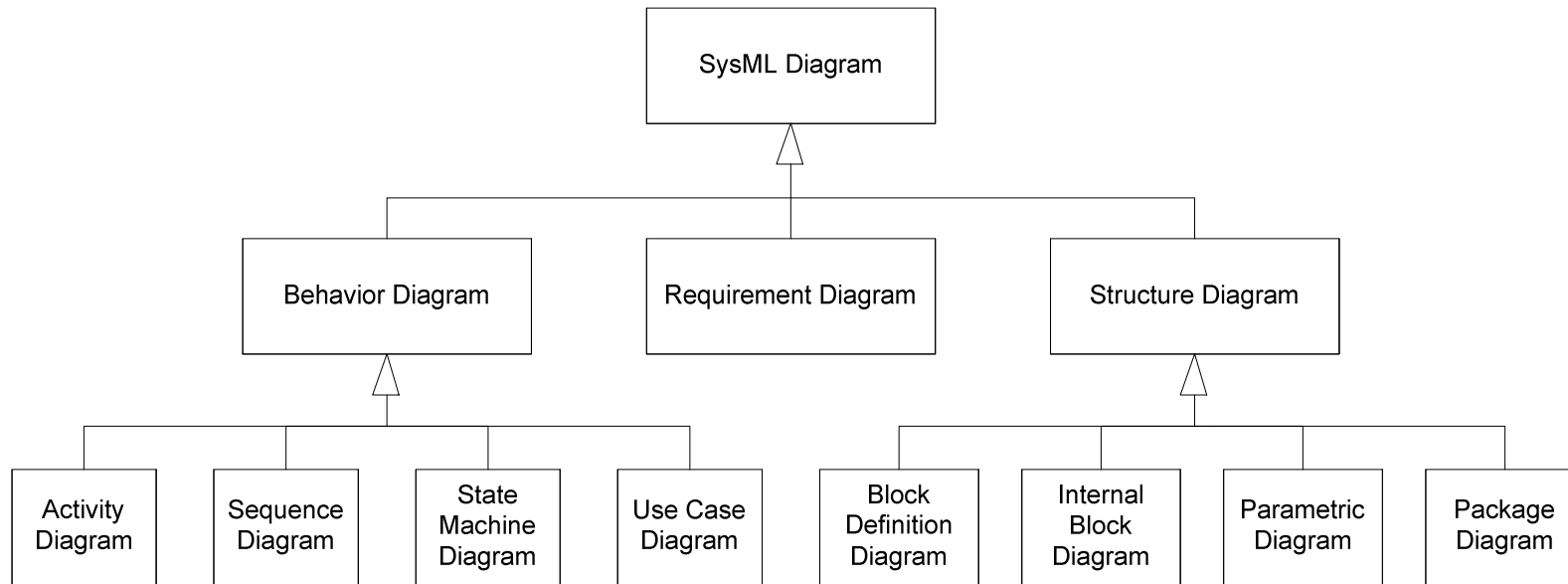# SysML Structural Diagrams 1
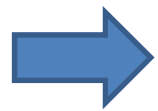
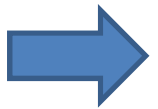## Introduction to Systems Engineering
I2ISE

SysML: Diagram types

Introduction

- There are 4 different types of structural diagams:
  - Block Definition Diagram (bdd) – Structural system elements called *blocks* and their composition
  - Internal Block Diagram (ibd) – Interconnection and interfaces between the *parts* of a block
  - Parametric diagram (par) – Constraints on property values
  - Package diagram (pkg) – The organization of a model into *packages* that contain model elements
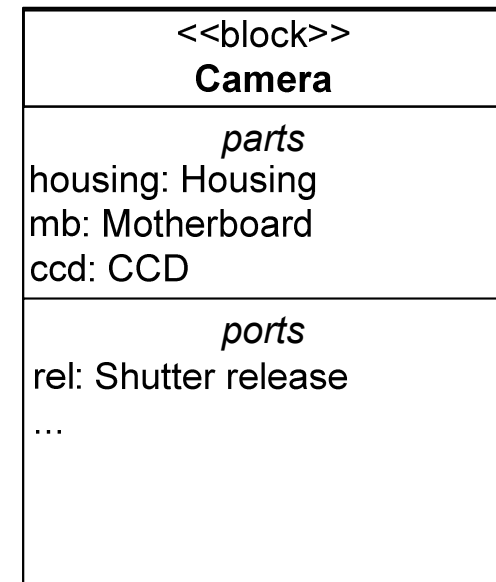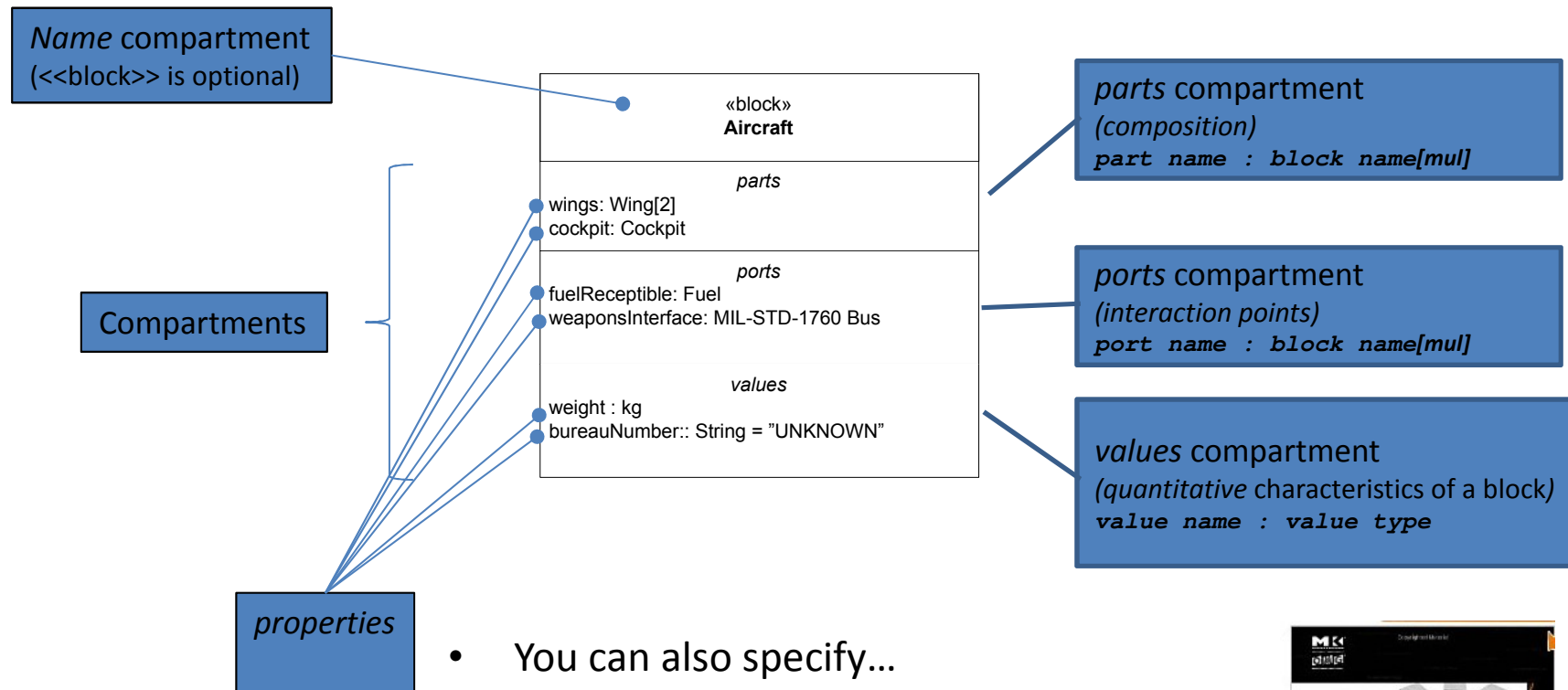
Blocks

SysML structural diagrams – the *blocks*

- The *block* is the fundamental model element for describing system structure
  - Hardware, software, person, facility, water, atmosphere, files,…

- The block is a *type*
  - A common description of similar *instances*, just like a C++ class

Blocks

- The block is drawn as a *rectangle* on a diagram canvas

- The block may be divided into *compartments*

- The top compartment always contains the block's *name*
    - *Name* is mandatory
    - `<<block>>` is optional

- Other compartments may be used to represent other block features
    - Parts, operations, ports, …

- Each compartment contains *properties*

<<block>>
Camera
parts housing: Housing mb: Motherboard ccd: CCD
ports rel: Shutter release …

Blocks – the works

Name compartment
(<<block>> is optional)

Compartments

properties

«block»
Aircraft

parts
wings: Wing[2]
cockpit: Cockpit

ports
fuelReceptible: Fuel
weaponsInterface: MIL-STD-1760 Bus

values
weight : kg
bureauNumber:: String = "UNKNOWN"

parts compartment
(composition)
`part name : block name[mul]`

ports compartment
(interaction points)
`port name : block name[mul]`

values compartment
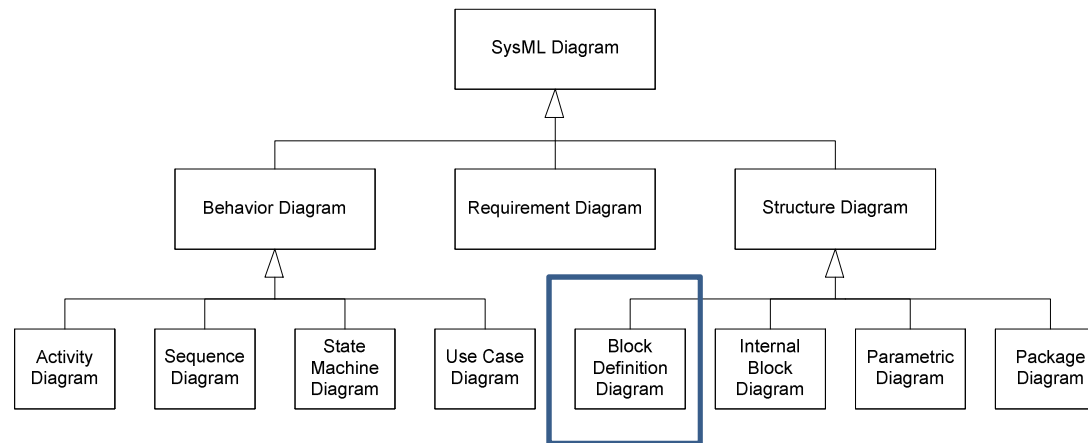*(quantitative characteristics of a block)*
`value name : value type`

- You can also specify…
  - *references* (weaker connections)
  - *value types* and their *units* and *dimensions*
  - *read-only properties*
  - *initial* property values, their *distribution*
  - *…*

A Practical Guide to SysML
The Systems Modeling Language

Sanford Friedenthal
Alan Moore
Rick Steiner
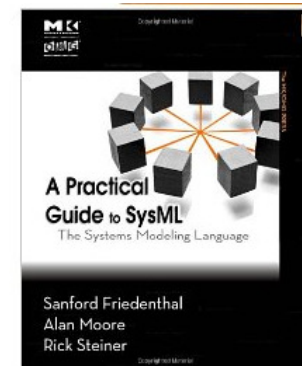
SysML
Block Definition Diagrams

```
                        ┌──────────────────┐
                        │  SysML Diagram   │
                        └──────────────────┘
                                 △
           ┌─────────────────────┼─────────────────────┐
  ┌──────────────────┐  ┌──────────────────┐  ┌──────────────────┐
  │ Behavior Diagram │  │ Requirement Diagram│  │ Structure Diagram│
  └──────────────────┘  └──────────────────┘  └──────────────────┘
           △                                            △
   ┌───────┼───────┬───────┐              ┌───────┬─────┼──────┬────────┐
┌────────┐┌────────┐┌────────┐┌────────┐┌────────┐┌────────┐┌────────┐┌────────┐
│Activity││Sequence││ State  ││Use Case││ Block  ││Internal││Parametric││Package│
│Diagram ││Diagram ││Machine ││Diagram ││Definition││Block ││ Diagram ││Diagram│
│        ││        ││Diagram ││        ││Diagram ││Diagram ││        ││        │
└────────┘└────────┘└────────┘└────────┘└────────┘└────────┘└────────┘└────────┘
```
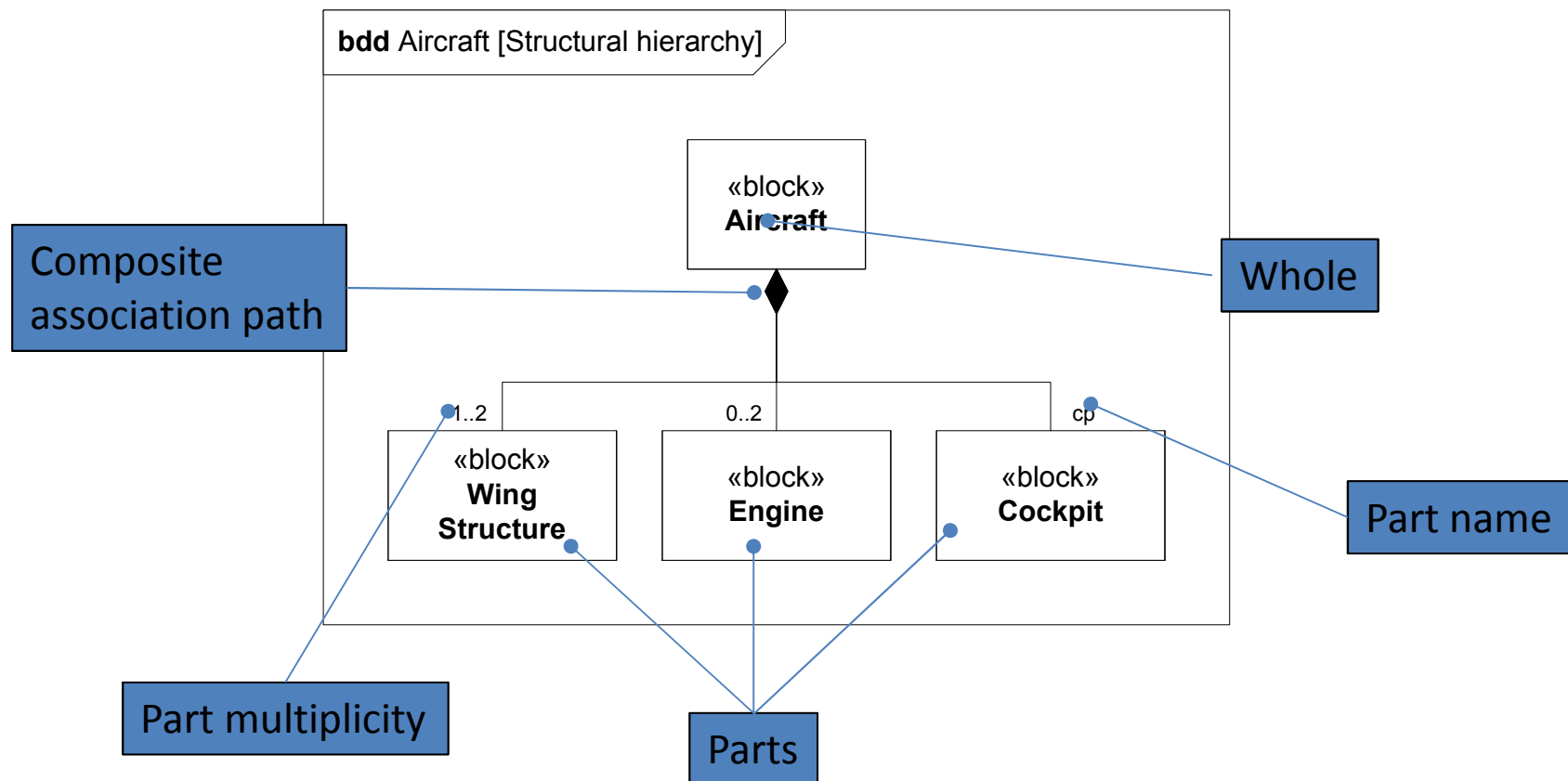
SysML: Block definition diagram

- A *Block Definition Diagram (BDD)* is used to define *blocks* and their relationship other blocks (their *composition*)

- A BDD may be used to define any kind of structure

  – Logical, physical, etc.

- BDDs are also used to define other relationships between blocks, e.g. allocation of functions to physical entities
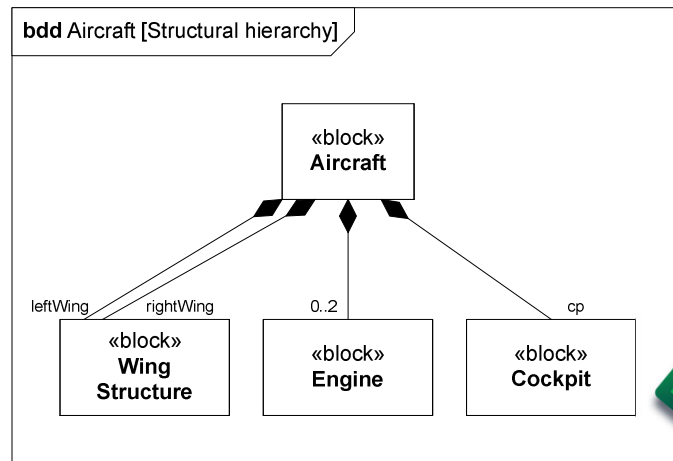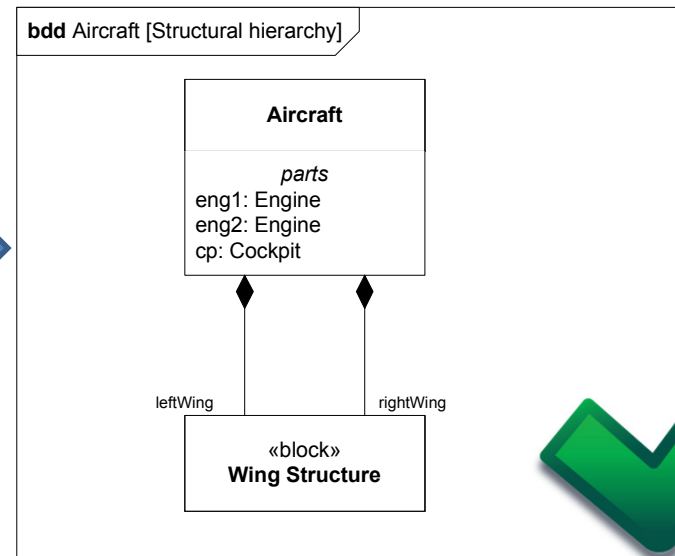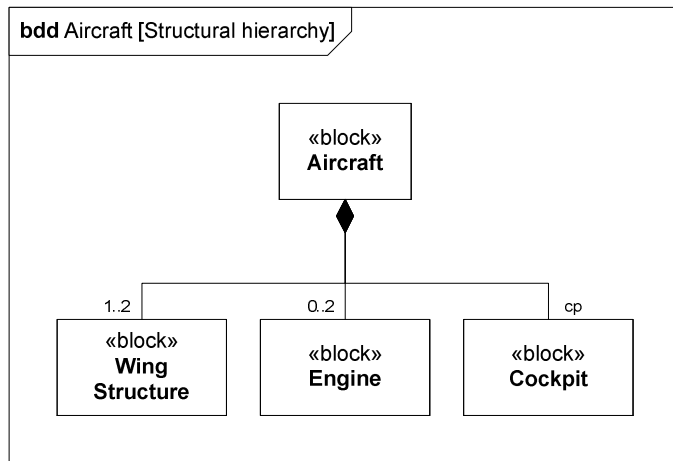
bdd: Composition relationships

The most common kind of relationship is *composition*:

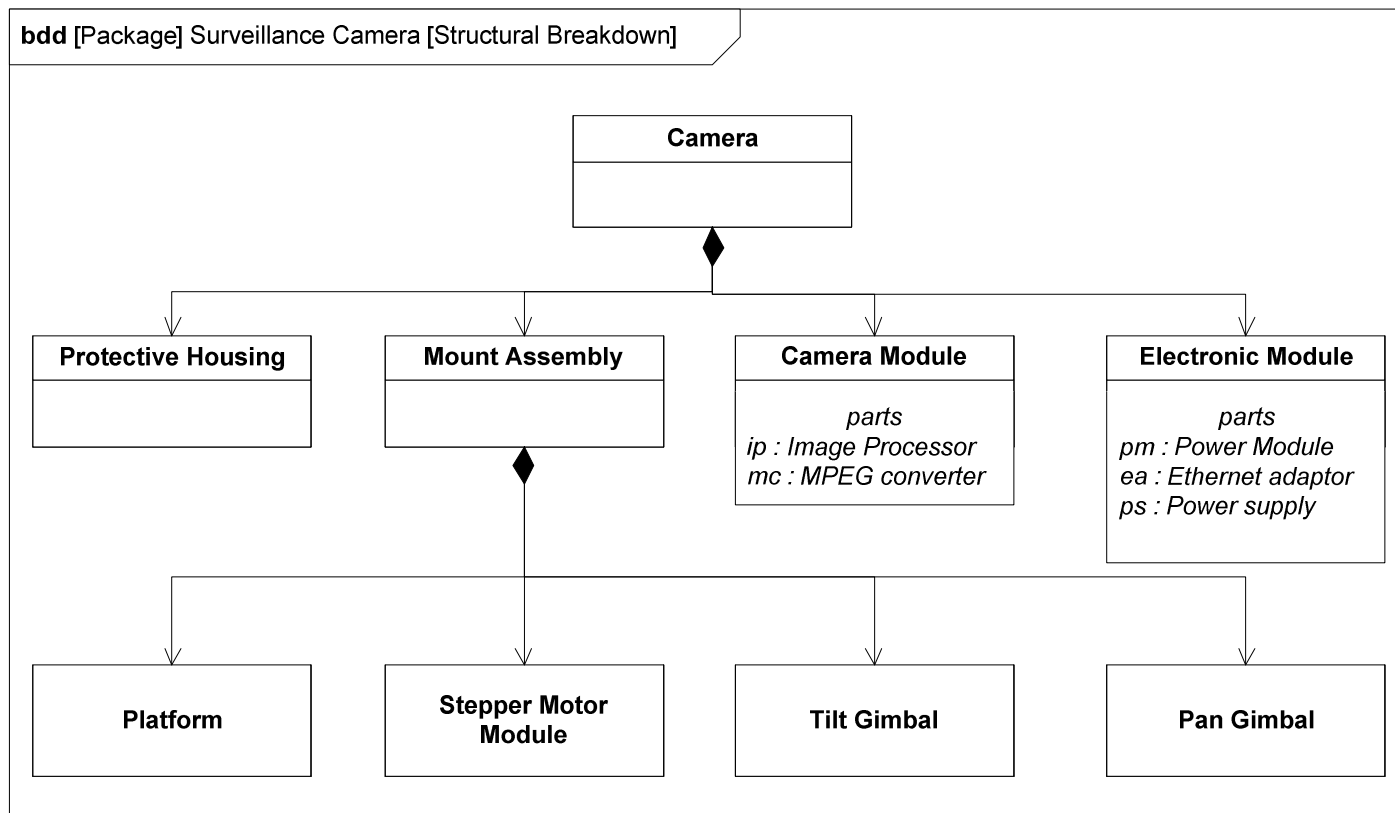- *"Consists-of"* or *"whole-part"* relationship, e.g. "an `Aircraft` consists-of 1-2 `wings`, 0-2 `engines` and 1 `cockpit`"

bdd: Variants

bdd Aircraft [Structural hierarchy]

«block»
Aircraft

1..2   0..2   cp

«block»
Wing Structure

«block»
Engine

«block»
Cockpit

bdd Aircraft [Structural hierarchy]

Aircraft

parts
eng1: Engine
eng2: Engine
cp: Cockpit

leftWing   rightWing

«block»
Wing Structure

bdd Aircraft [Structural hierarchy]

«block»
Aircraft

leftWing   rightWing   0..2   cp

«block»
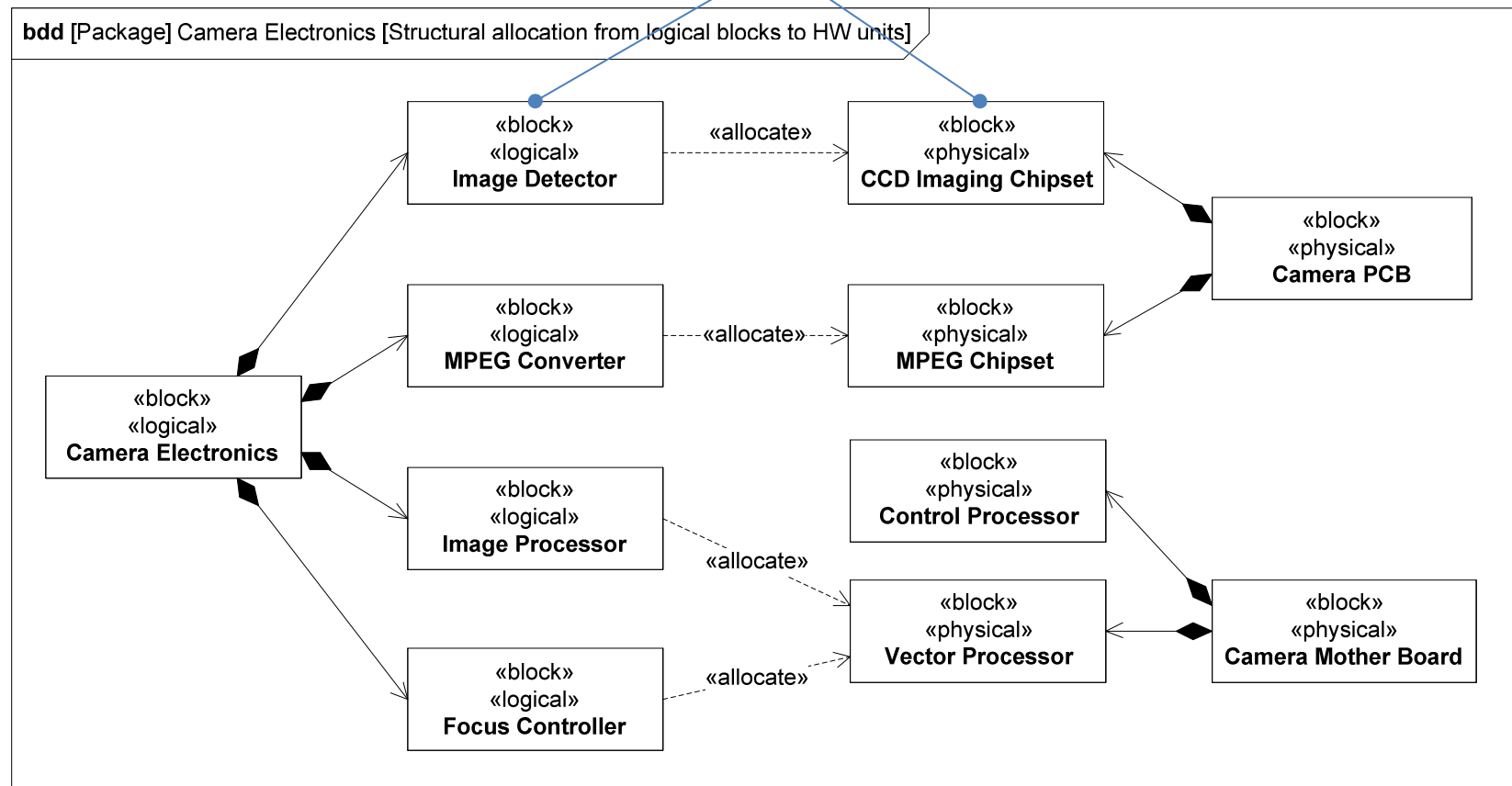**Wing Structure**

«block»
Engine

«block»
Cockpit

bdd: Deeper hierarchy

How would you read this diagram? *"A camera consists of…"*

bdd: Another use

Logical functions *allocate* physical components

bdd [Package] Camera Electronics [Structural allocation from logical blocks to HW units]

«block»
«logical»
Camera Electronics

«block»
«logical»
Image Detector

«allocate»

«block»
«physical»
CCD Imaging Chipset

«block»
«logical»
MPEG Converter

«allocate»

«block»
«physical»
MPEG Chipset

«block»
«physical»
Camera PCB

«block»
«logical»
Image Processor

«block»
«physical»
Control Processor

«allocate»

«block»
«logical»
Focus Controller

«allocate»

«block»
«physical»
Vector Processor

«block»
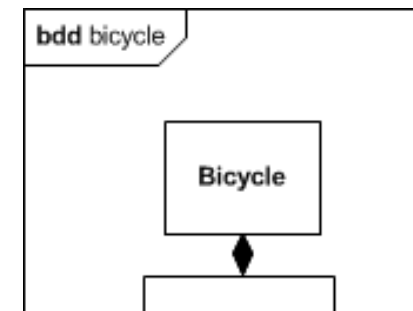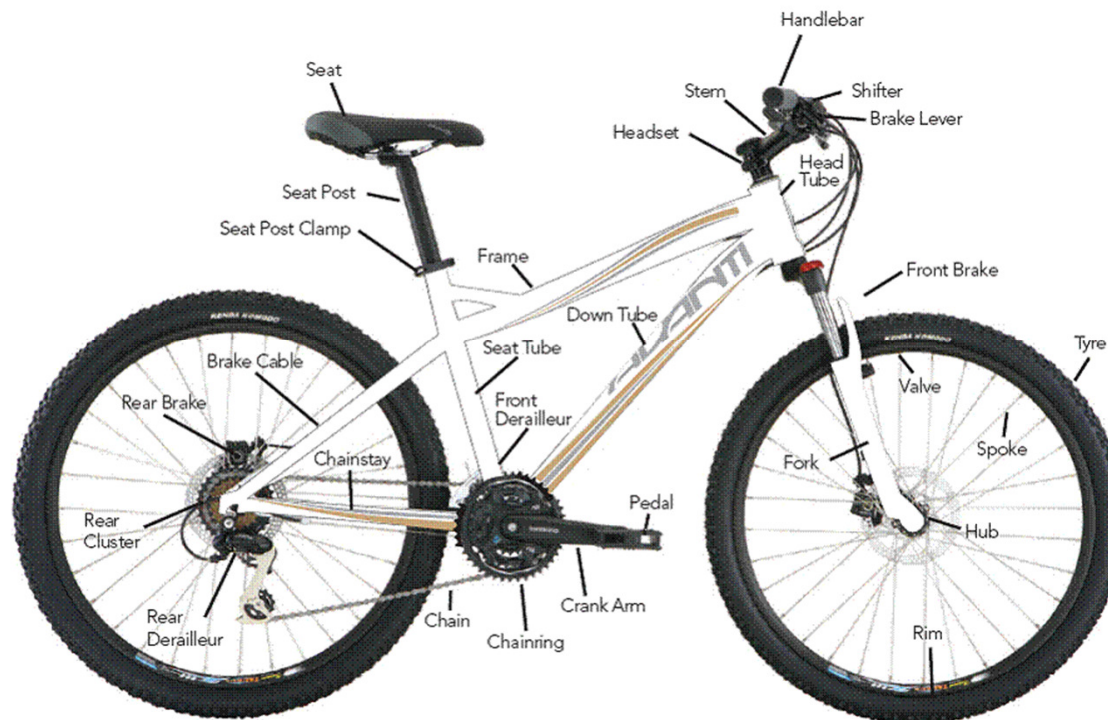«physical»
Camera Mother Board

bdd: Defining the system's context

- A top-level bdd is often used to define the context of the system of interest (SOI)
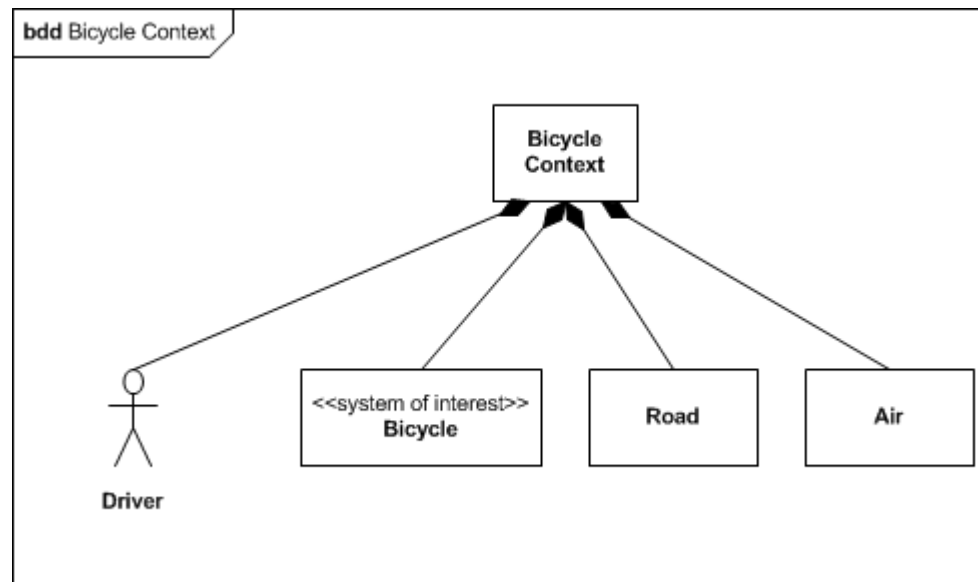
Your turn!

- Describe the hierarchical structure of a bicycle in a Block Definition Diagram (bdd).
    - The bdd must contain at least 2 hierarchical levels.
    - Use *Composite Association Paths* to relate the blocks.

Your turn!

- Create a bdd for an access control system