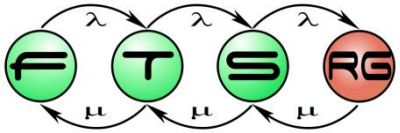
Modeling Structure with Blocks in SysML

Critical Embedded Systems

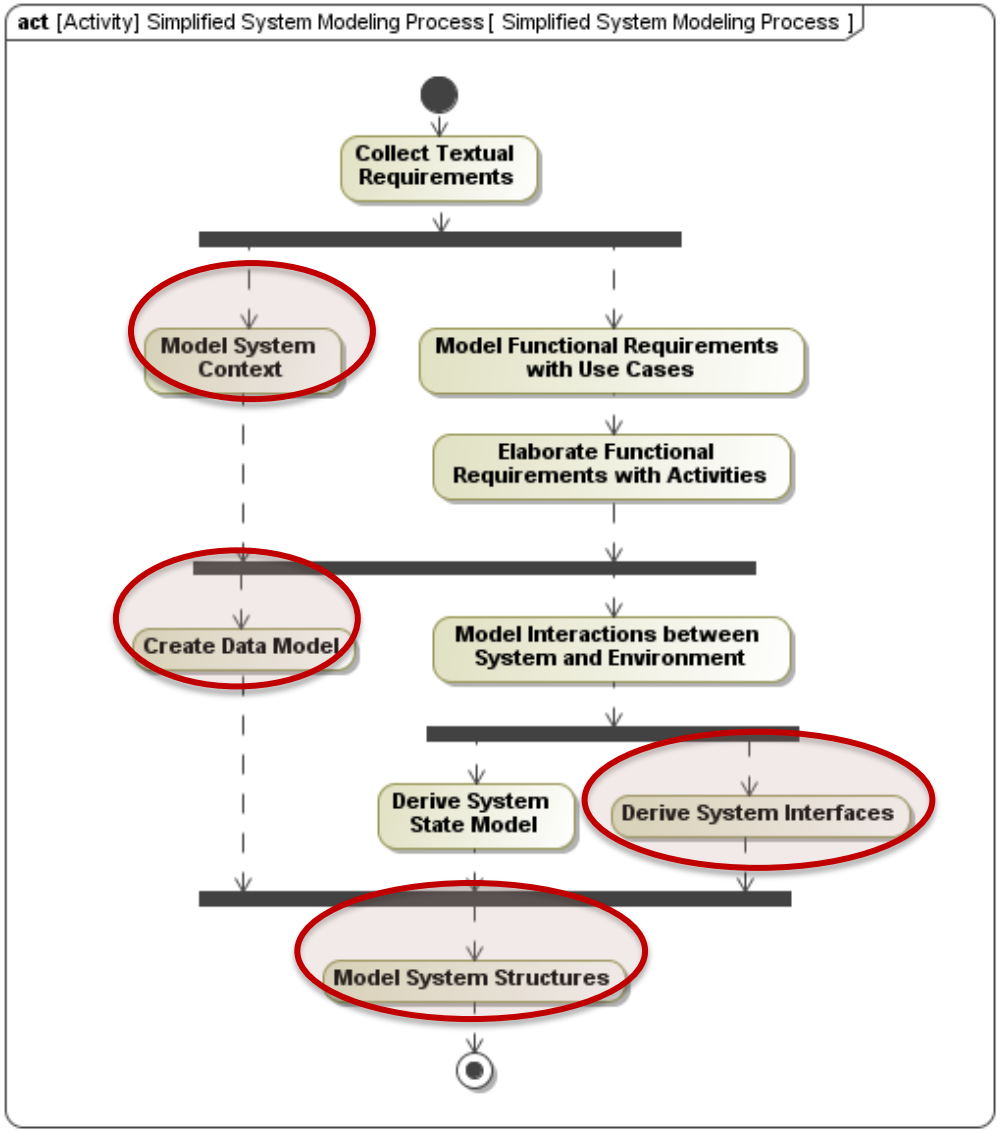
Gábor Guta, PhD

Prepared by
Budapest University of Technology and Economics
Faculty of Electrical Engineering and Informatics
Dept. of Measurement and Information Systems



This material can only used by participants of the course.

System Modeling Process



Block Definitions

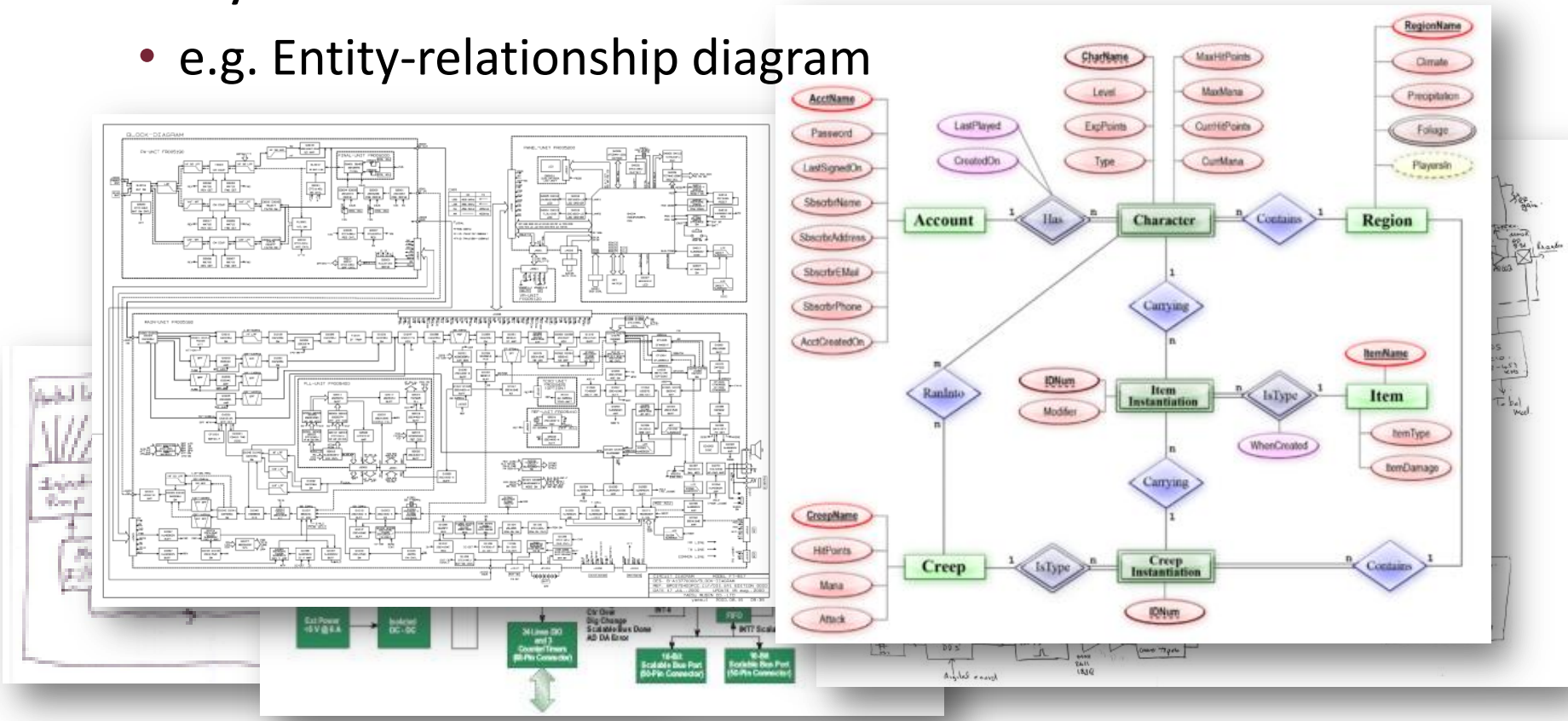
Block Definition Diagrams

What is it about?

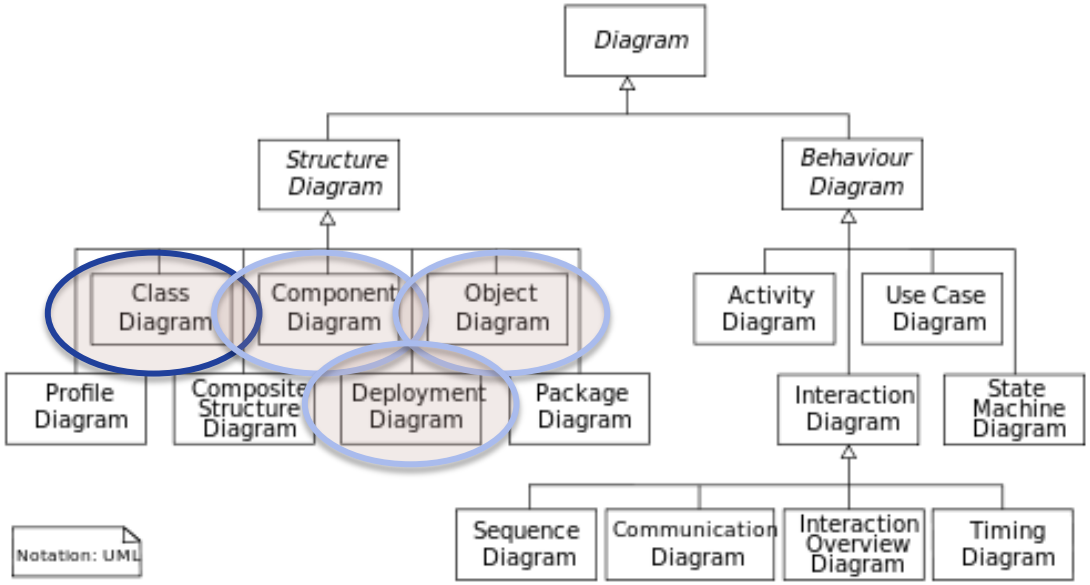
Context of the Modeling Aspect

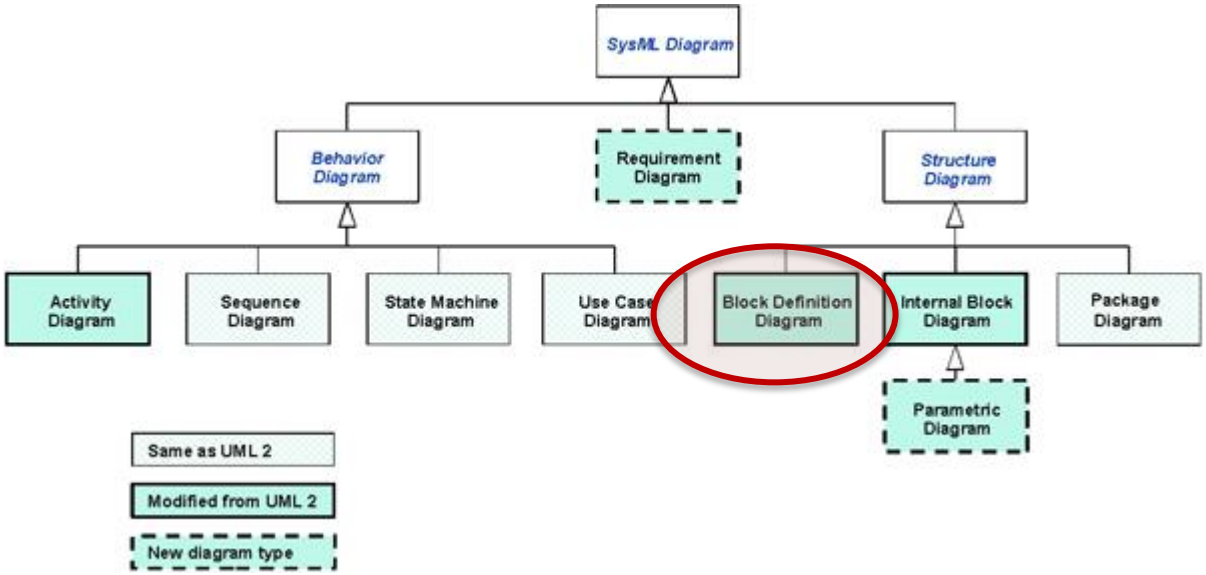
Roots & Relations

- Engineers draws blocks from the beginning
  - By hand or with CAD tools
  - Many formats
    - e.g. Entity-relationship diagram



Block Definition Diagram

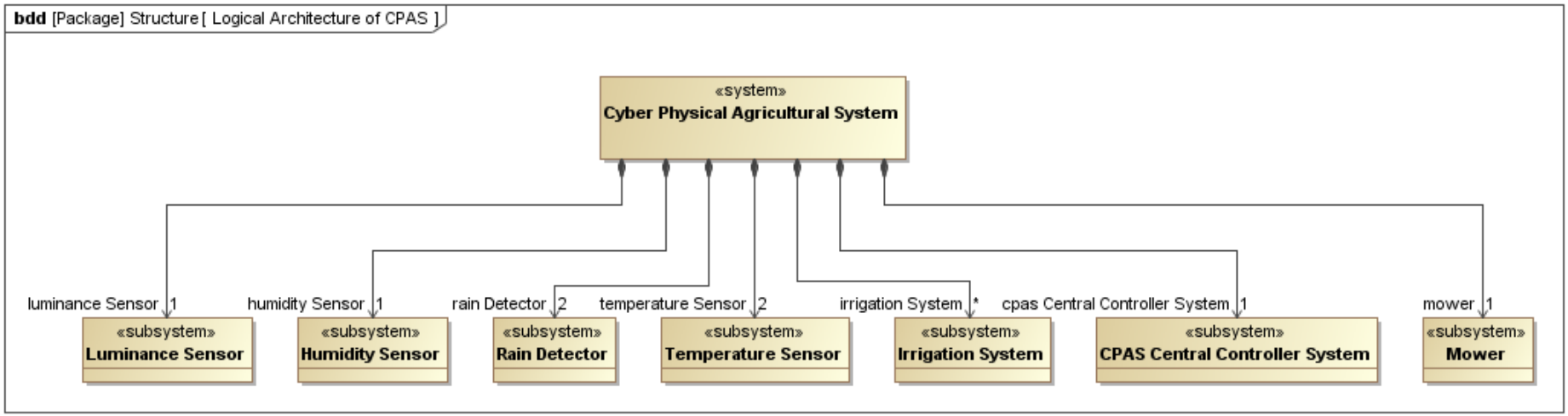




Modeling Aspect

What are the building blocks?

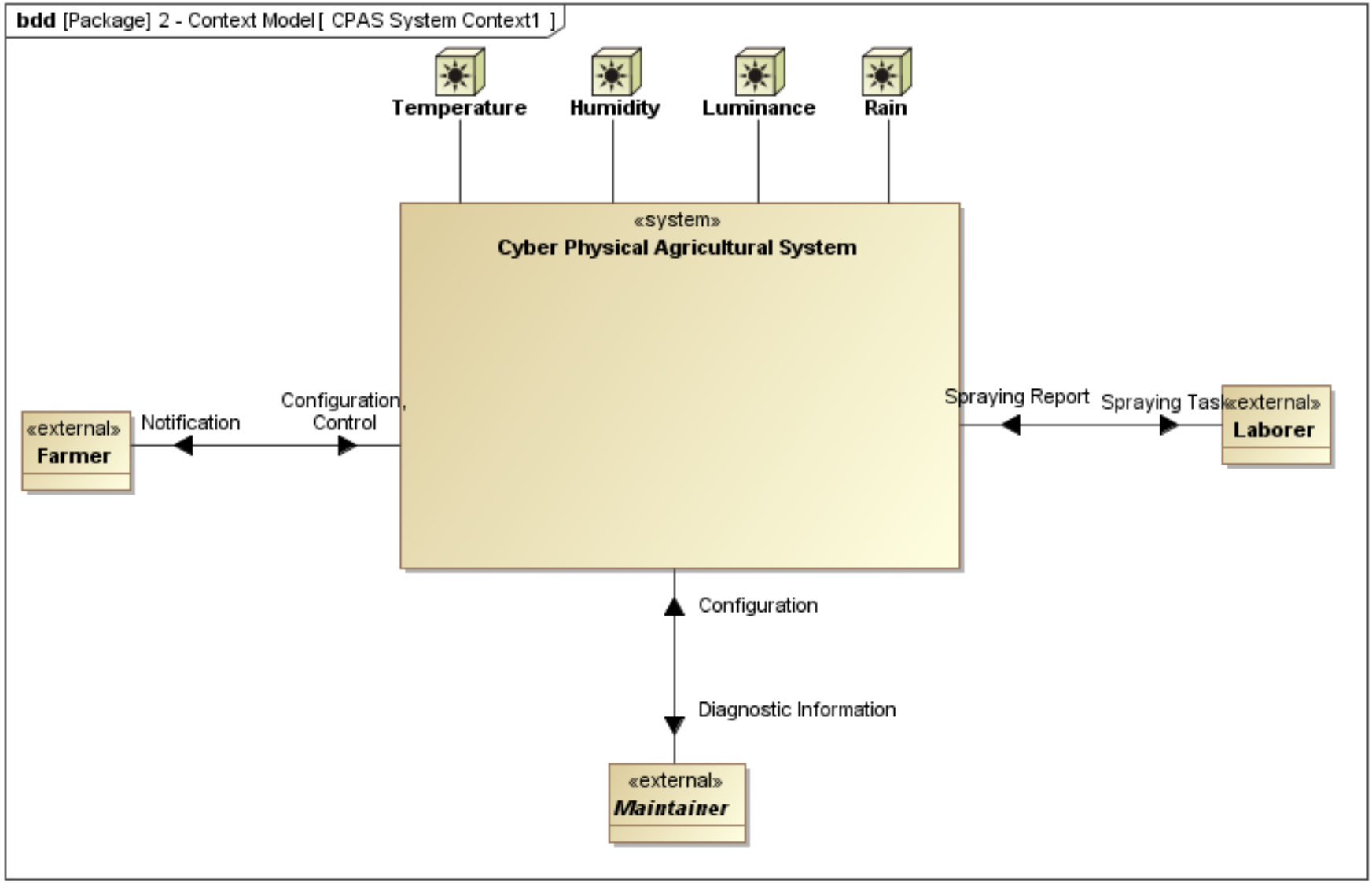
What are their relations in general?



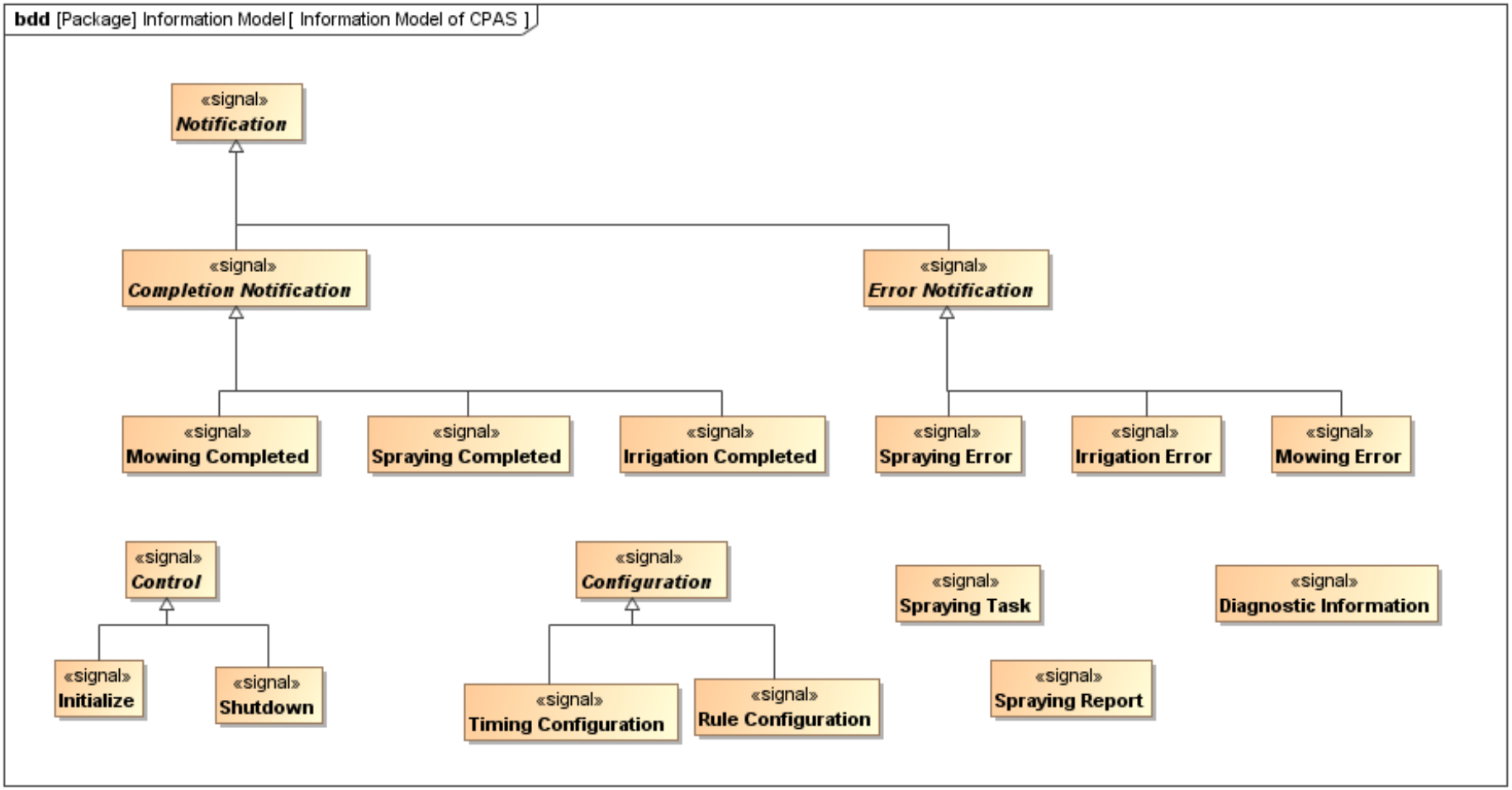
Objectives

- Define component types
  - Support organization into taxonomy (generalization)
- Define data model
- Define system decomposition
- Define interfaces and ports

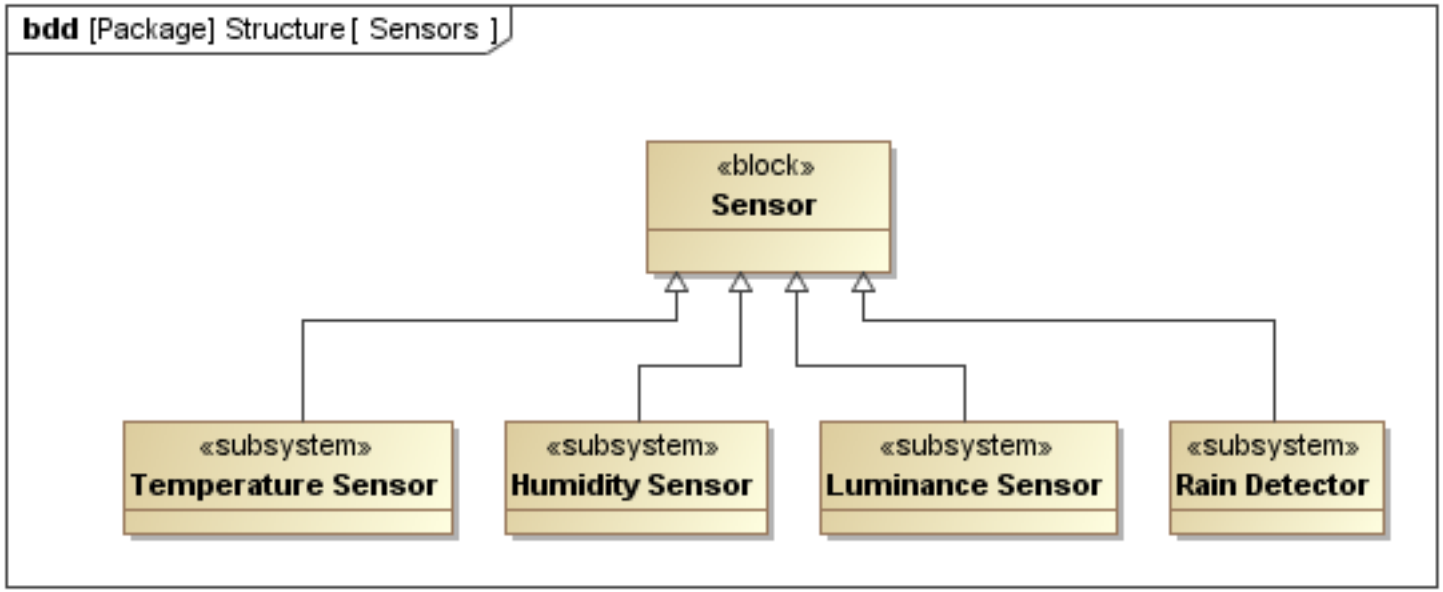
Example – System Context



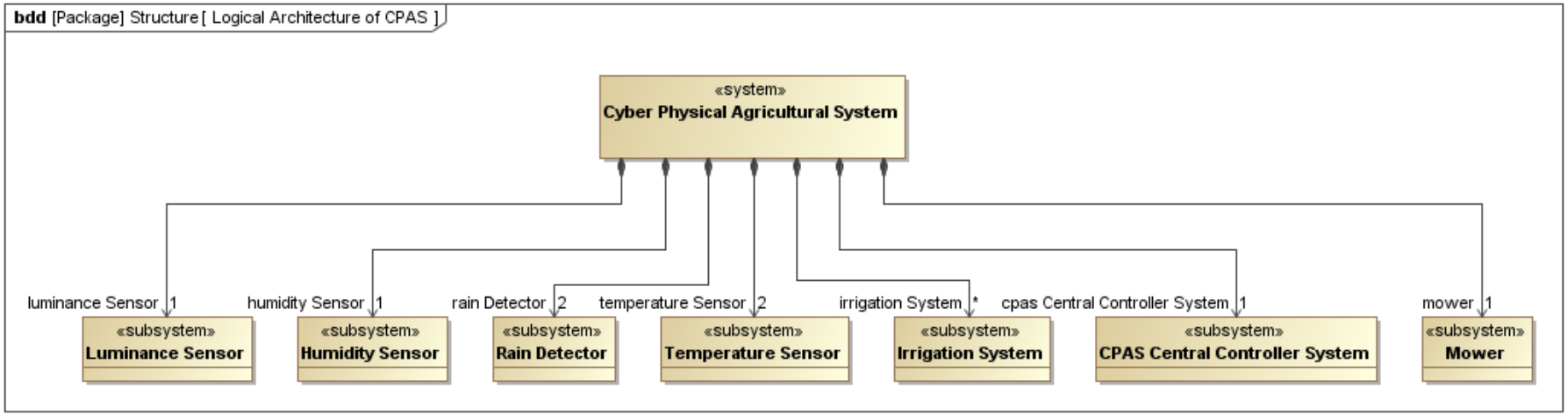
Example - Signals



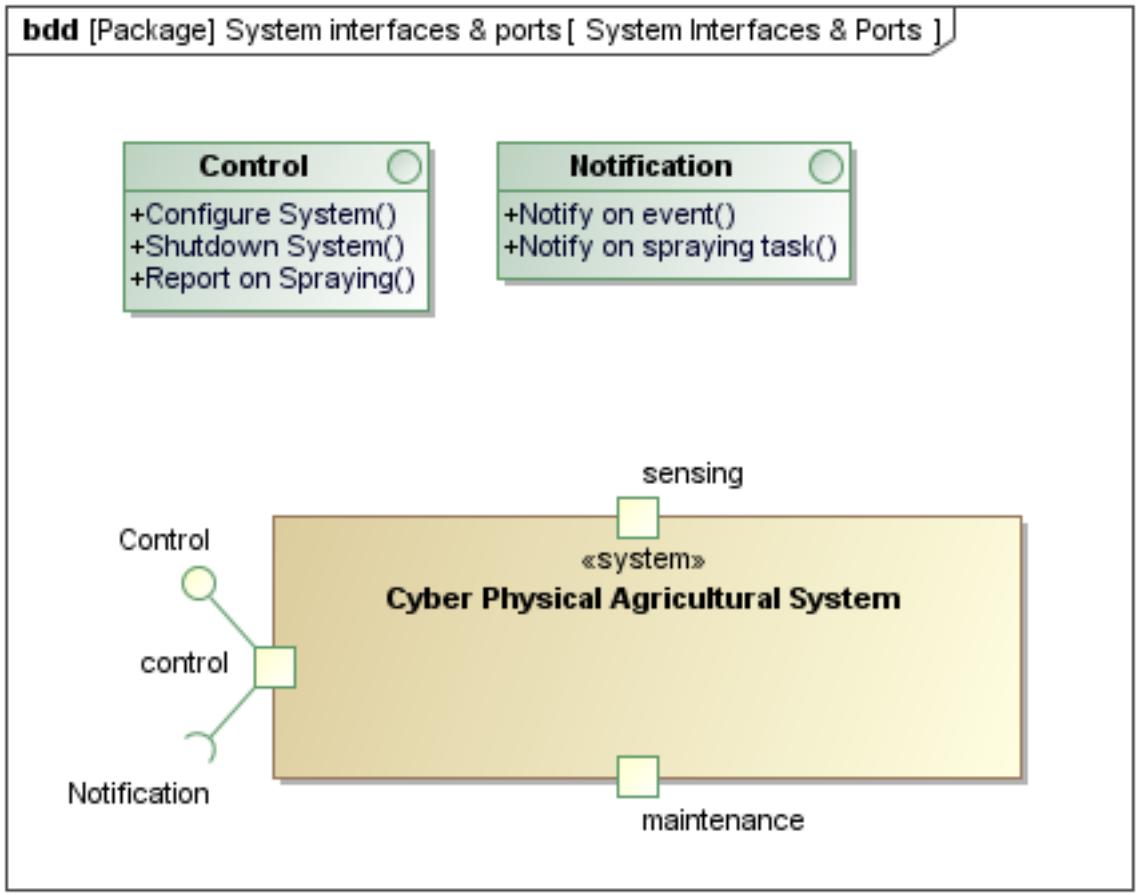
Example – Component definition



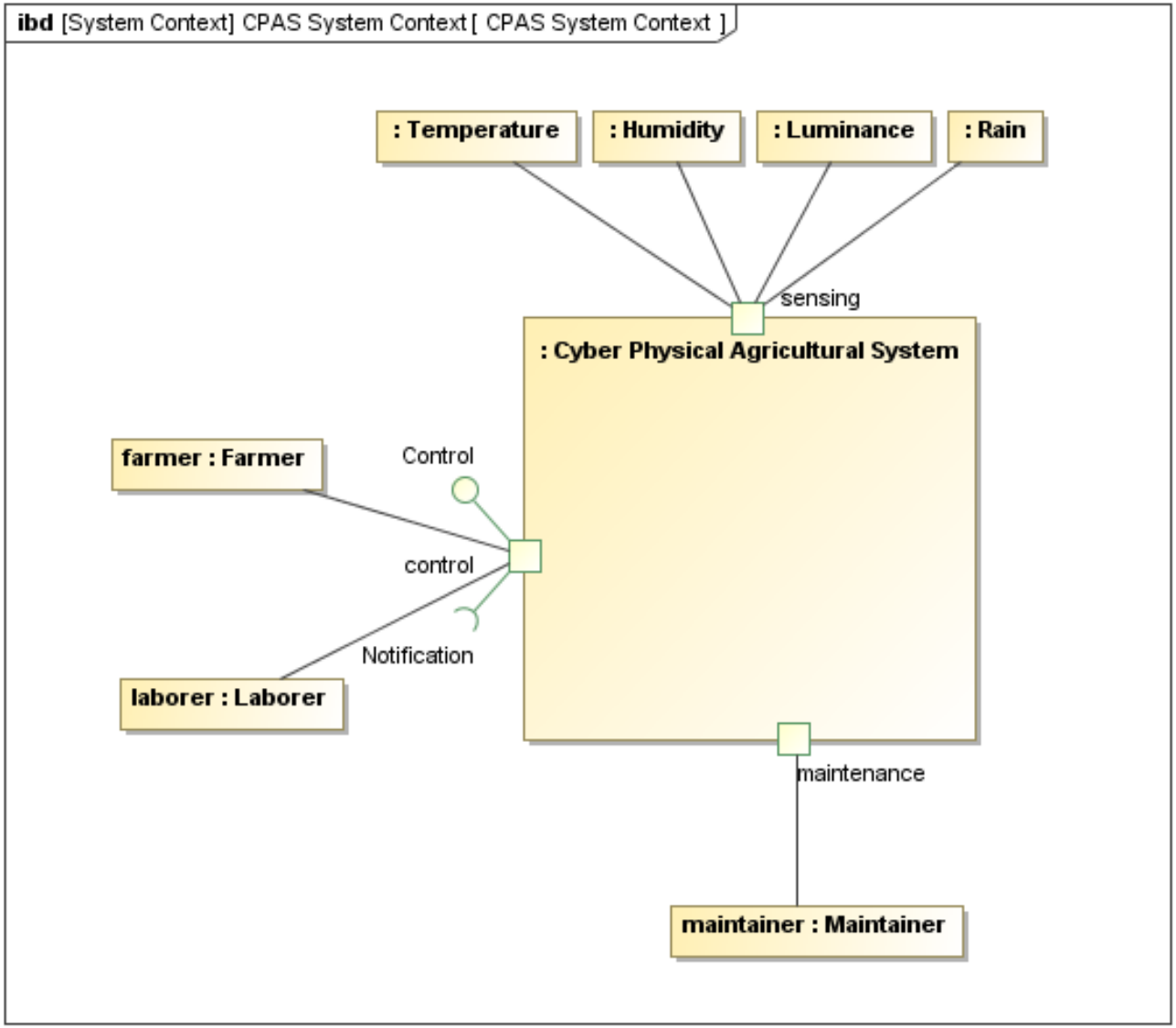
Example – System Decomposition



Example – Interfaces and Ports



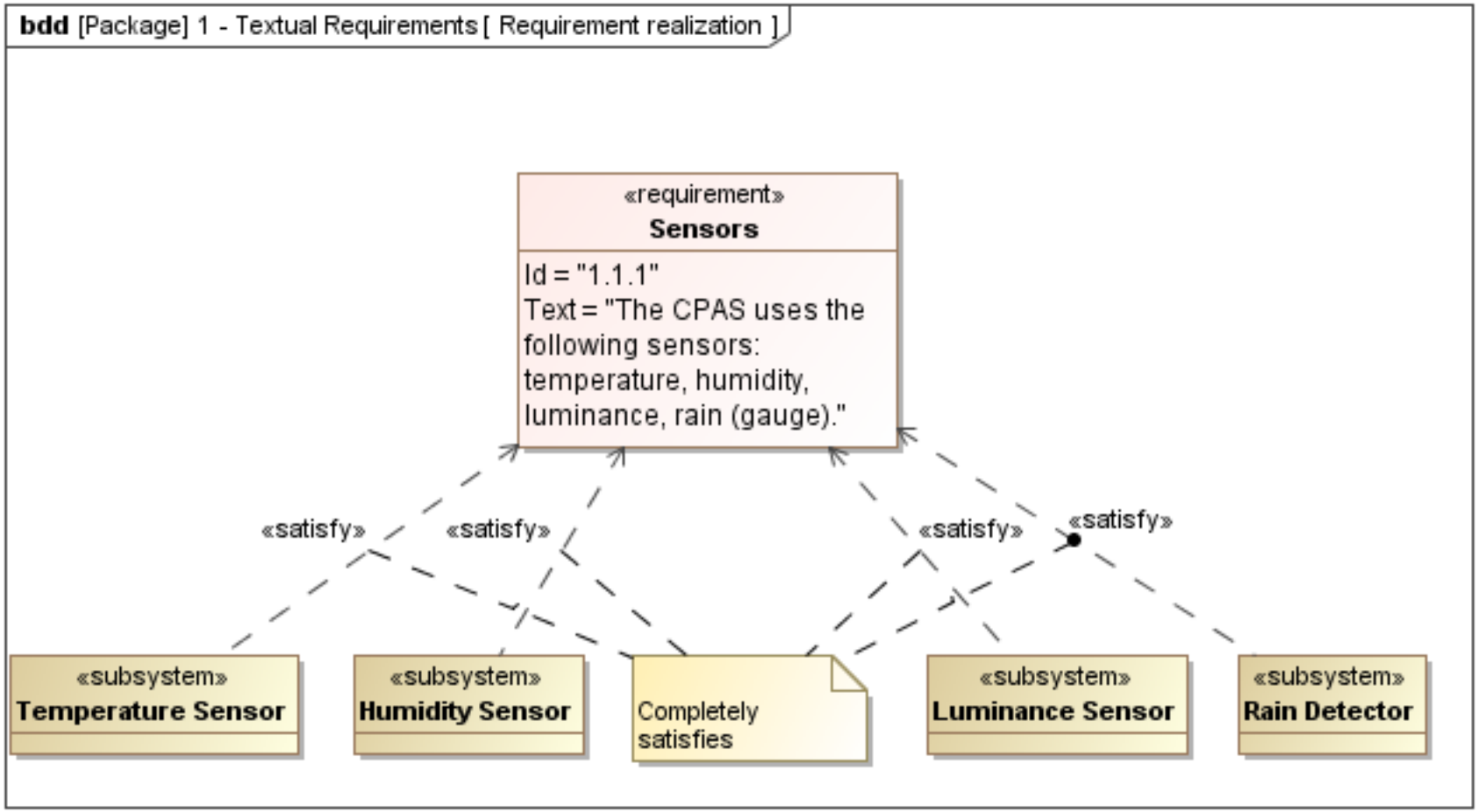
Example – System Context with Ports



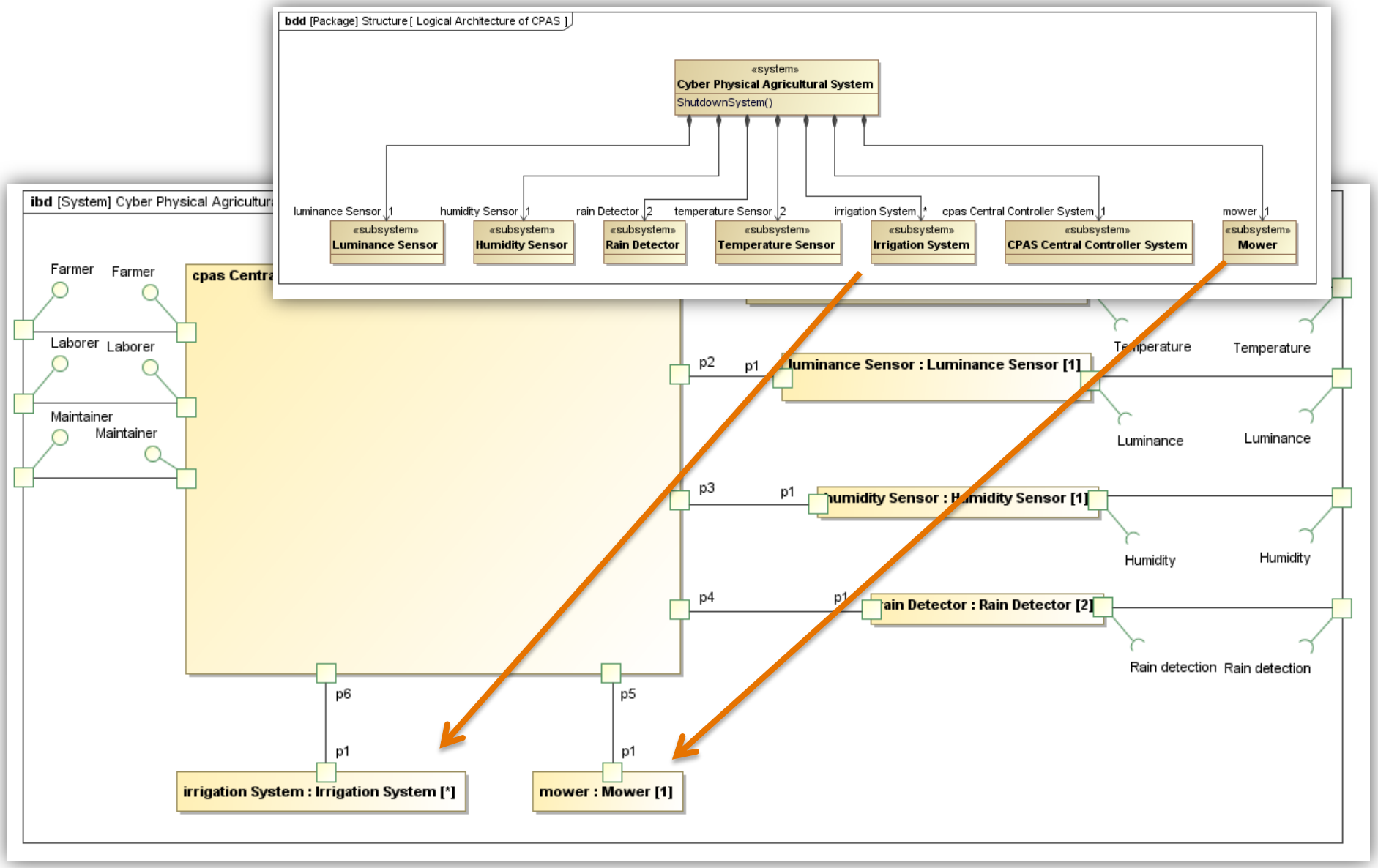
Relations to other aspects

- Realizes requirements
- Provides types for parts & ports
- Executes actions
- Defines participants in collaborations
- Provides context for state machines

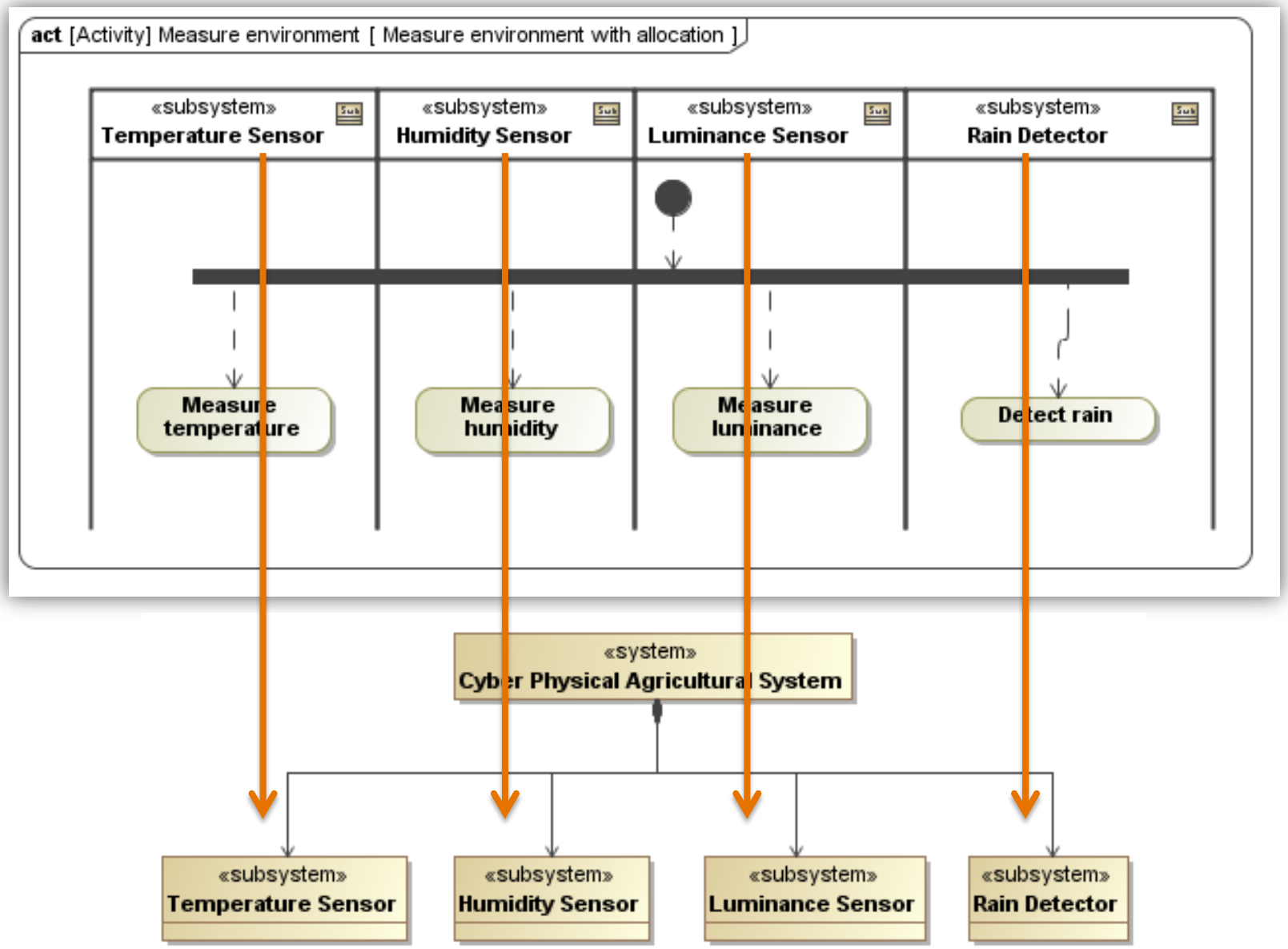
Realizes requirements



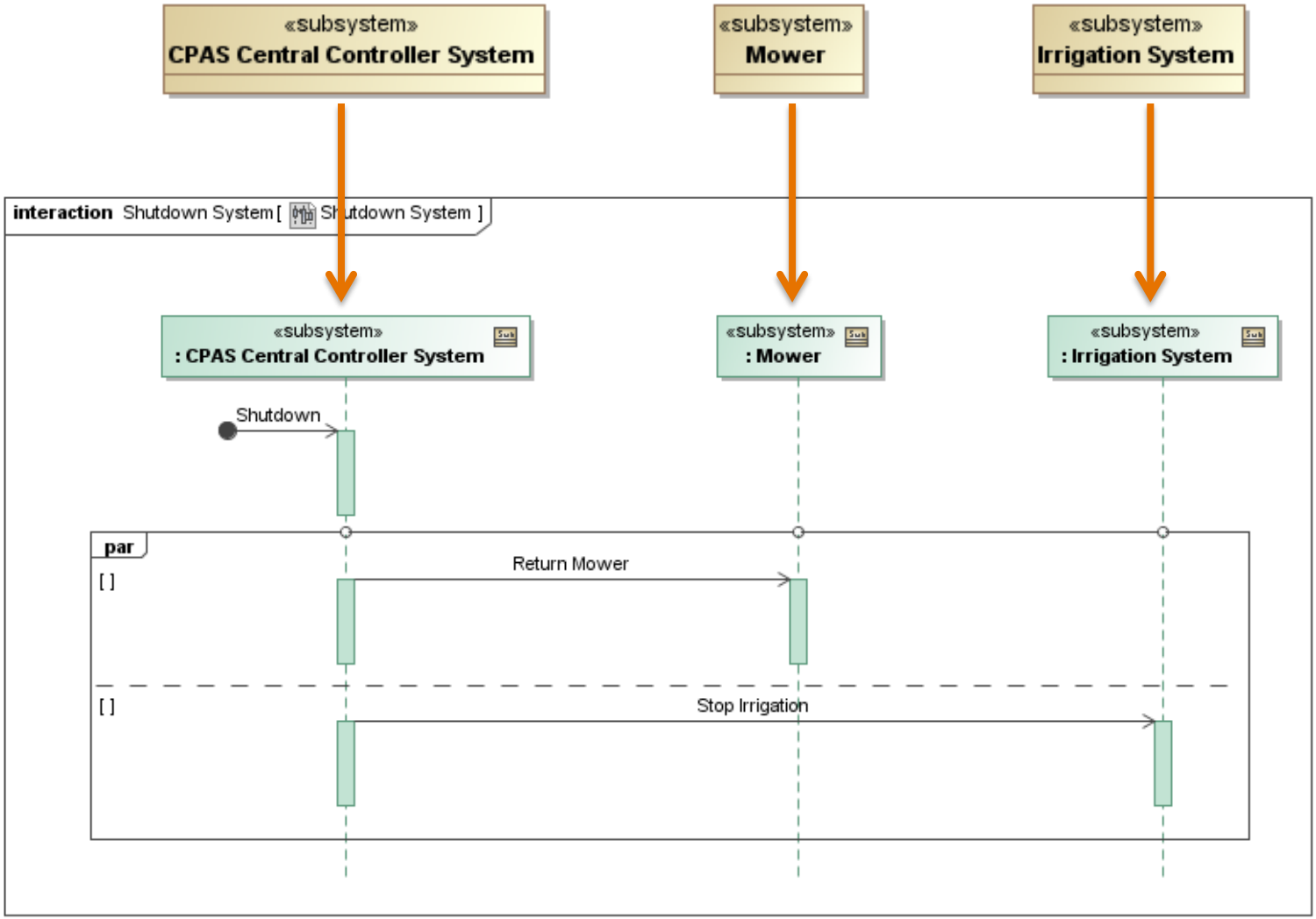
Provides types for parts & ports



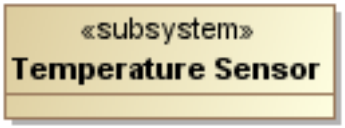
Executes actions

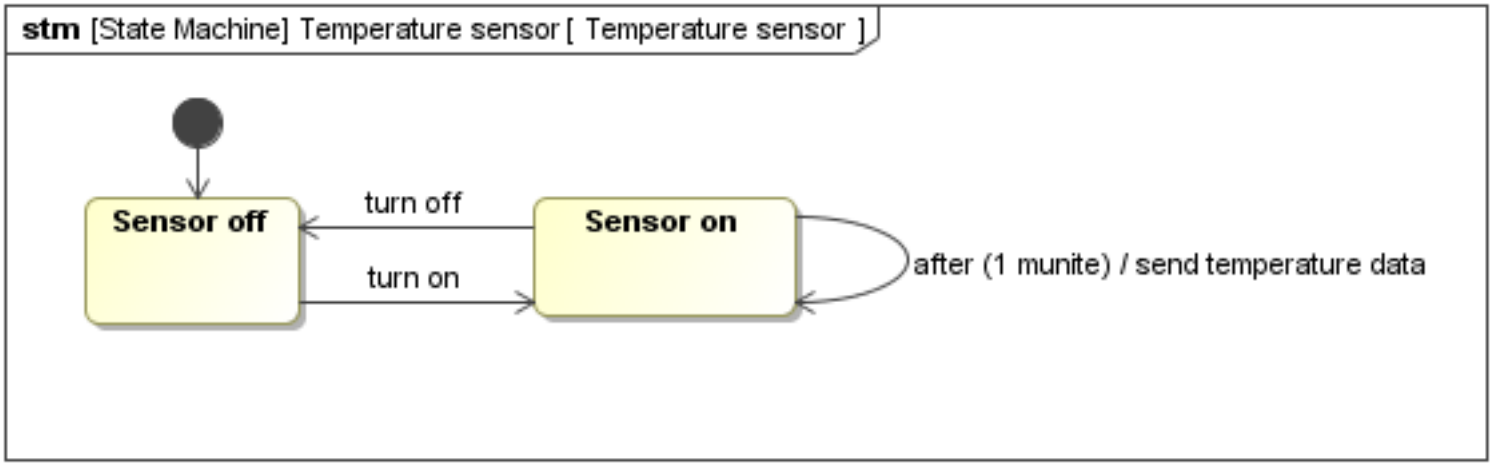


Defines participants in collaborations



Provides context for state machines





What are the building blocks?

Modeling Elements & Notation

Essential Elements of Block Definition Diagrams

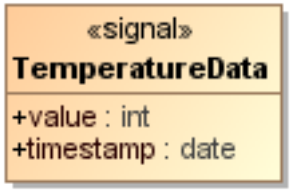
- Nodes
  - Block nodes
  - Signals
  - Value Type, Quantity Kind and Unit
  - Enumeration nodes
  - Actor nodes
- Paths
- Ports and Flows
- Constraint blocks

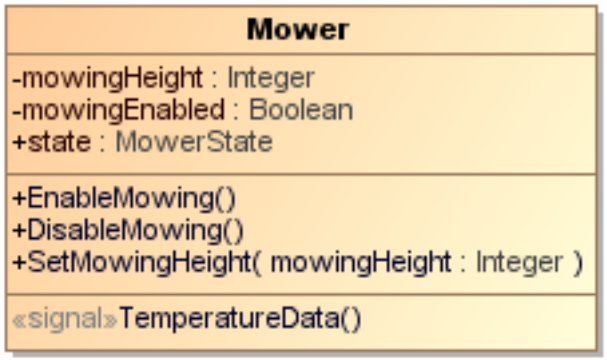
Block nodes

- Basic structural elements, that describe the structure of the system
- Compartments
  - Property types (e.g. parts, properties, references, values)
  - Behavioral (operations, receptions)
  - Constraints
- Describe
  - (Sub)Systems – Hardware / Software / Data
  - Person

Signal

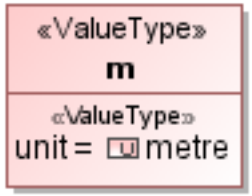
- A signal is a *specification of send request instances* communicated between objects.
- The receiving object handles the received request instances as specified by its *receptions*.
- The data carried by a send request (which was passed to it by the send invocation occurrence that caused that request) are represented as *attributes of the signal*.
- A signal is defined independently of the classifiers handling the signal occurrence.

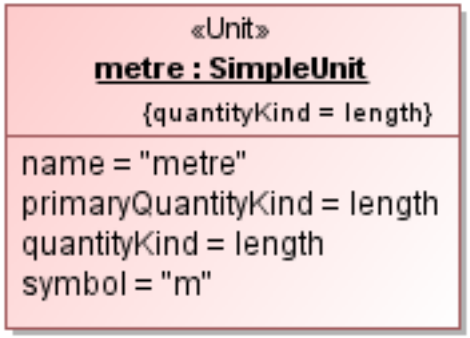


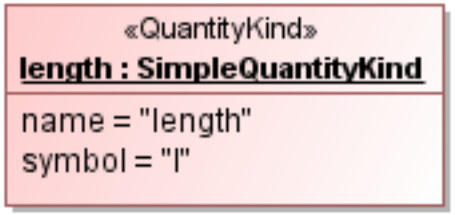


Value Type, Quantity Kind and Unit

- Uniform definition of a quantity
- Value Type
  - Data type, that can have Unit and QuantityKind nodes
  - Type for value properties in blocks
- Quantity Kind
  - Identifies a physical quantity
- Unit
  - Describes the structure of a physical unit – the unit of measure
  - Must be related to a Quantity Kind

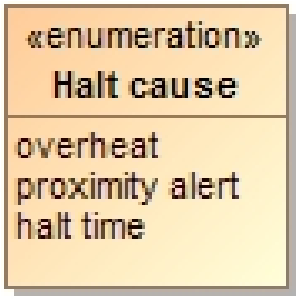






Enumeration and Actor nodes

- Enumeration
  - Defines a type, the value range of which is a limited set of named values, called literals.



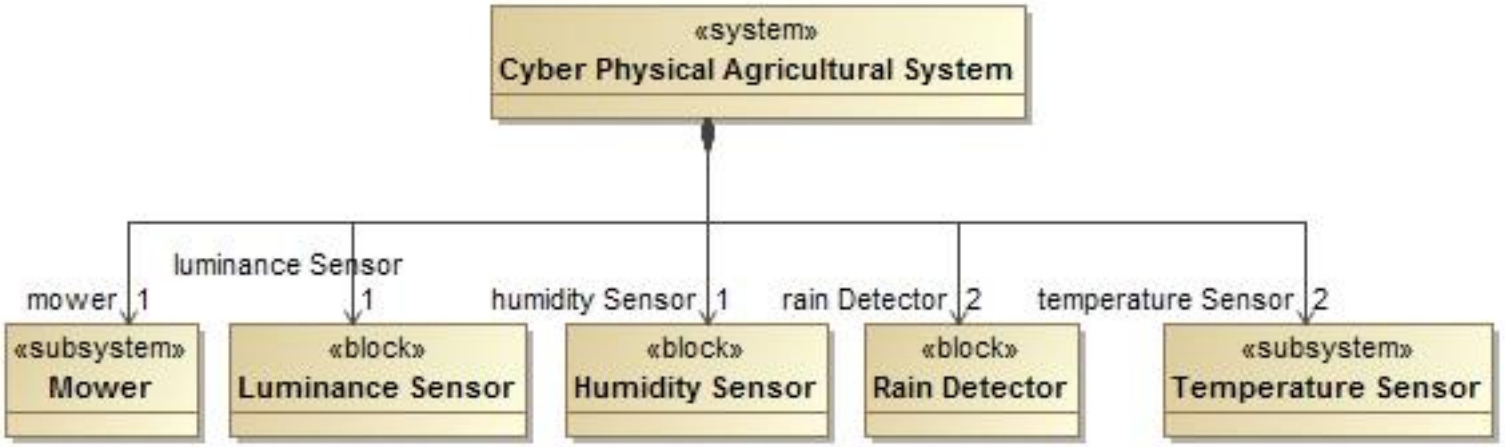
- Actor
  - Represents any stakeholder (human, organisation or external system) that participates in the use of the system

Defining Paths between Blocks

- Paths
  - Part Association
  - Shared Association
  - Reference Association
  - Association Class
  - Generalization
  - Dependency

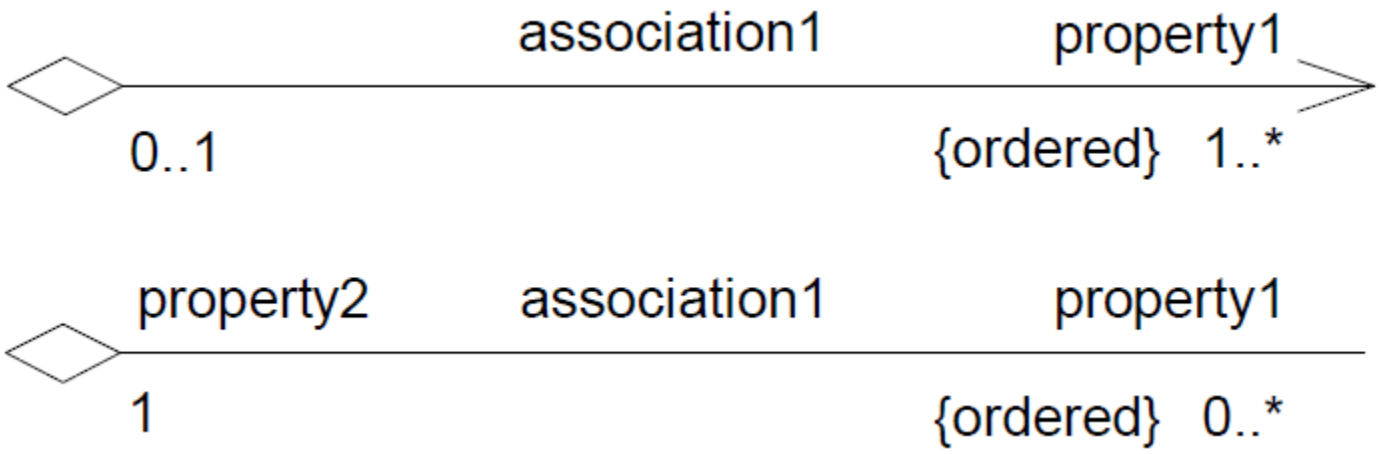
Part Association

- Specifies a strong whole-part hierarchy
  - From a *composite*
  - To a *composite part*
- Denoted with a black diamond on the whole end
- Role name on the part end
- Can be directed or undirected



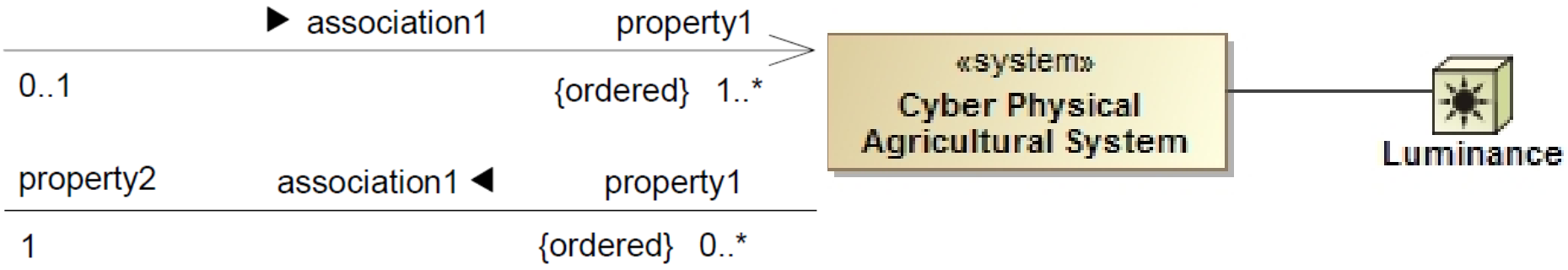
Shared Association

- Specifies a weaker whole-part hierarchy
  - "Shared" indicates, that the whole part is not the only one, it can be more of it
  - The parts are not owned by the whole part
- Denoted with an empty diamond



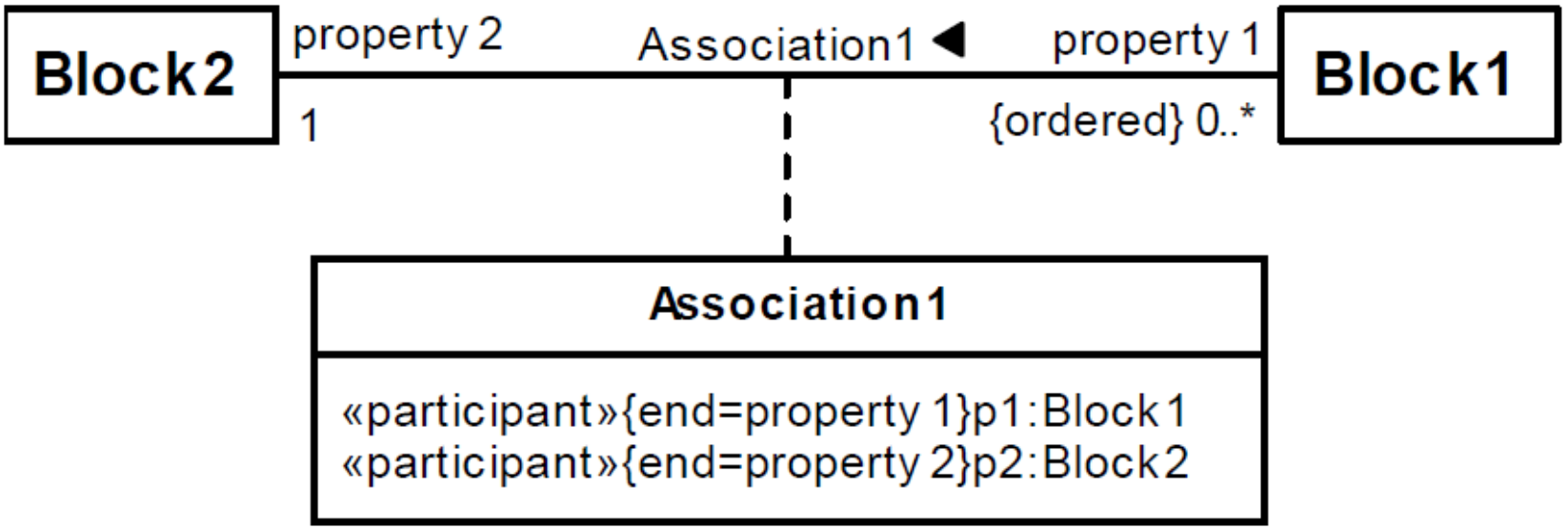
Reference Association

- Represents a relationship between two blocks
  - Undirected: reference in both blocks
  - Directed: reference only in one block
- Can have properties
  - Multiplicity
  - Name
  - Reference on both sides



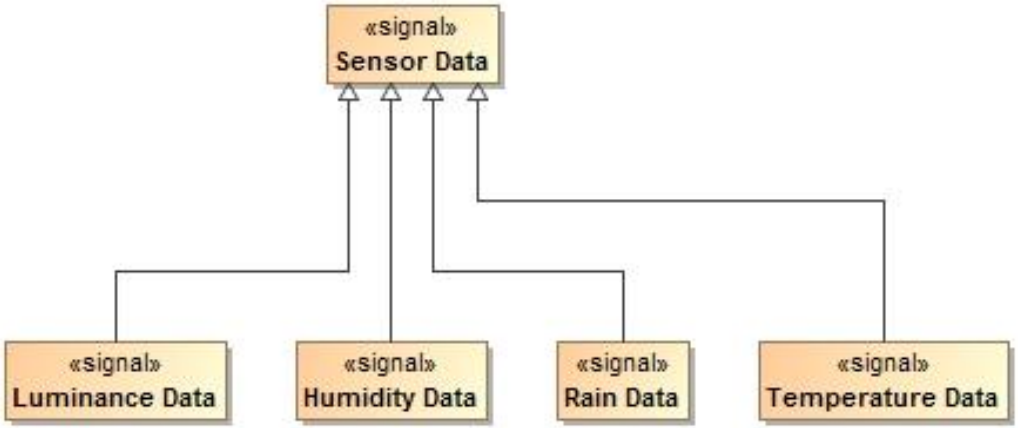
Association Class

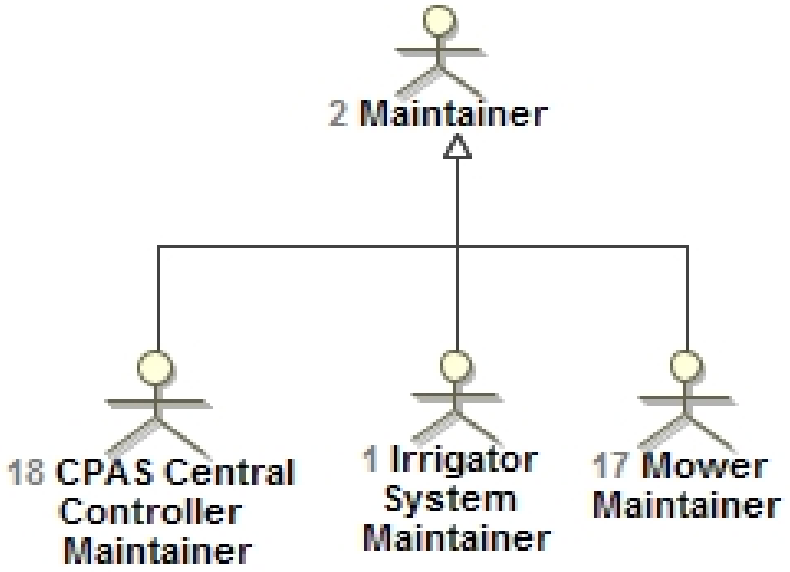
- Describes the structural properties of an association
- Combination of
  - Association
  - Block



Generalization

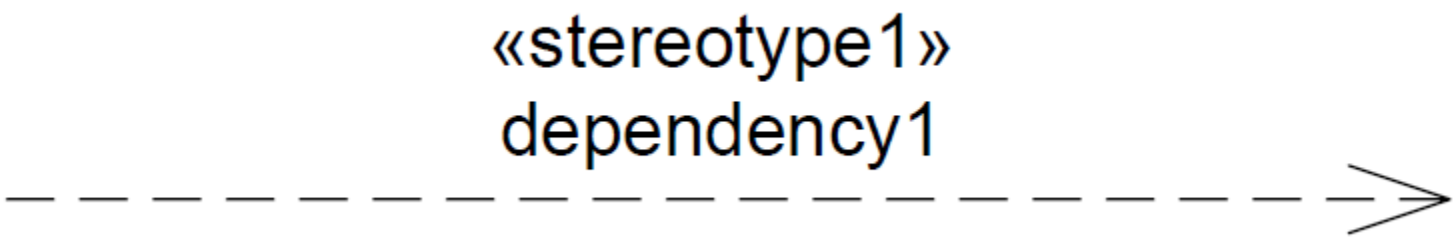
- Specifies an object oriented relationship between a more general block and a more specific one (ISA relationship)
- Denoted with a closed arrowhead from the specific block to the general one





Dependency

- Between two elements
- One element needs the other element for its
  - Specification
  - Implementation
- Almost between any model elements

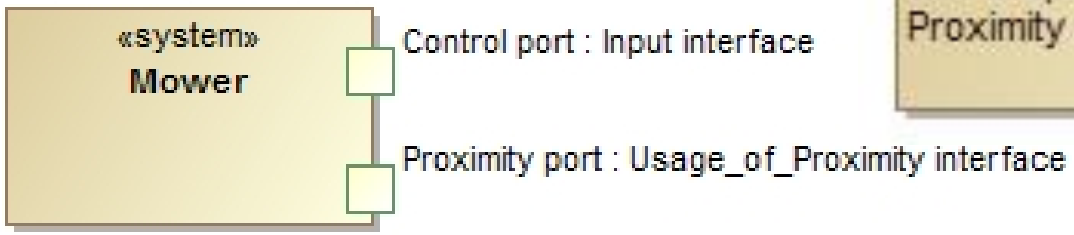


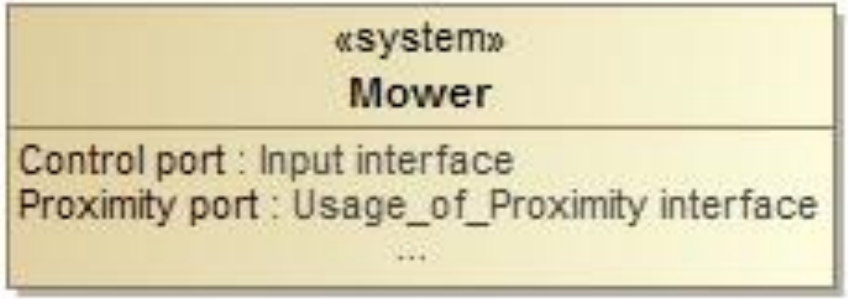
Defining Ports and Flows

- Port
- Flow Property
- Interface

Port

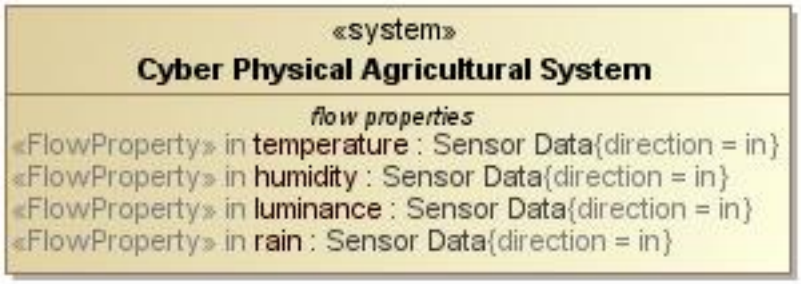
- Port
  - Interaction points at which external entities can be connected
  - Limits and differentiates the possible connection types
  - Defines the available features (e.g. properties, operations)
  - More denotation alternative





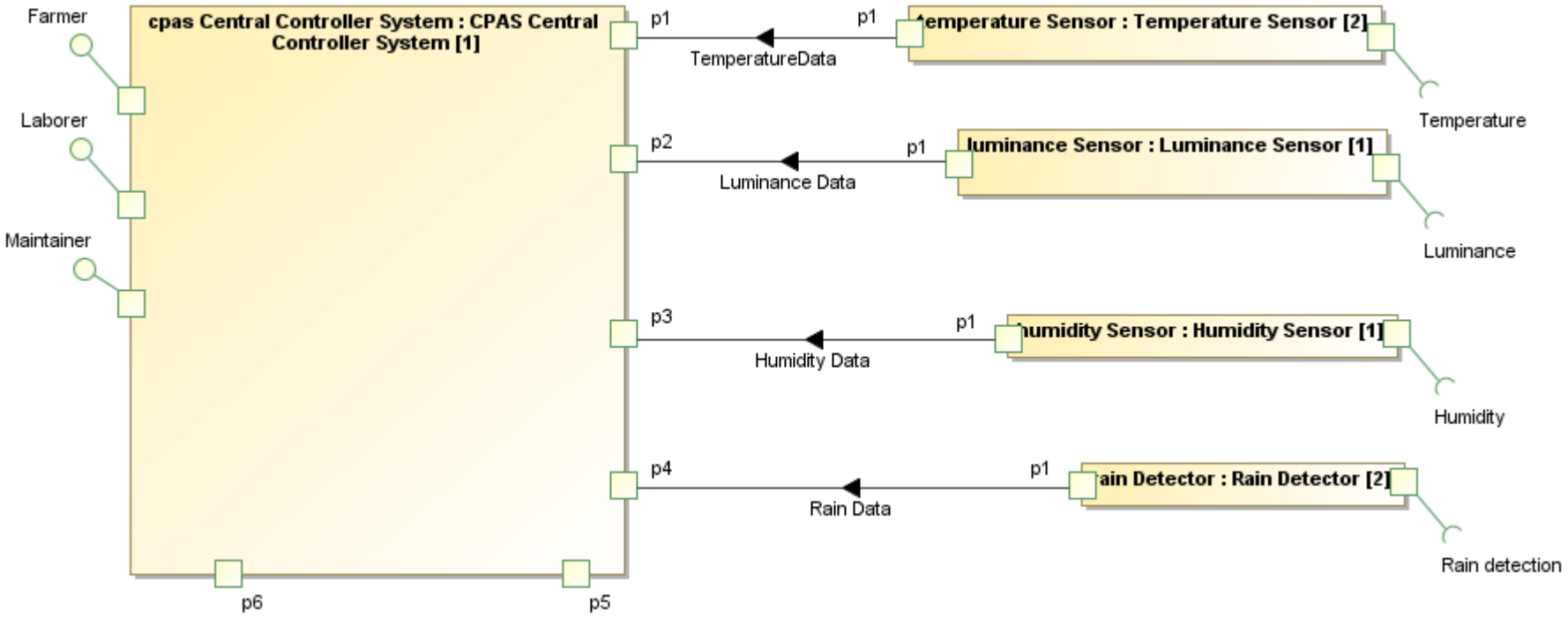
Flow Property

- Specifies the possible types of flowing items on a port
  - Part of flow specification
  - *What "can" flow?*
    - Data
    - Material
    - Energy
    - …



Flow Item

- Specifies what flows between the blocks
  - *What "does" flow?*

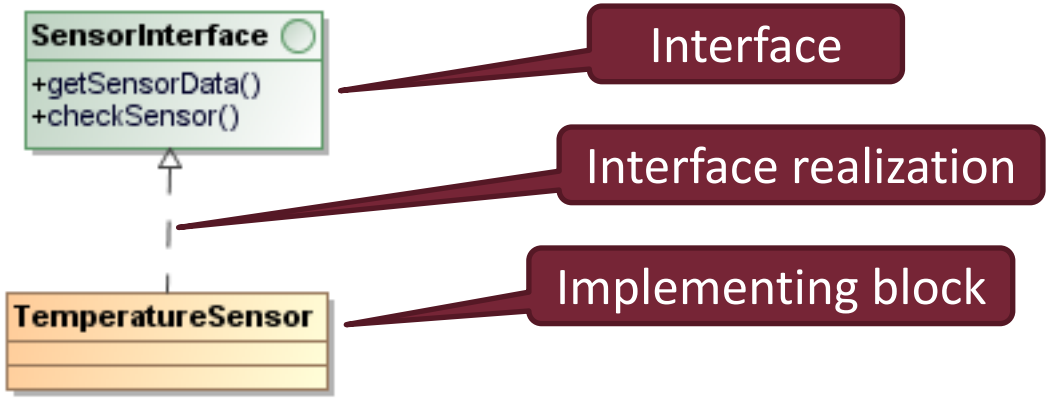


Interfaces

Definition

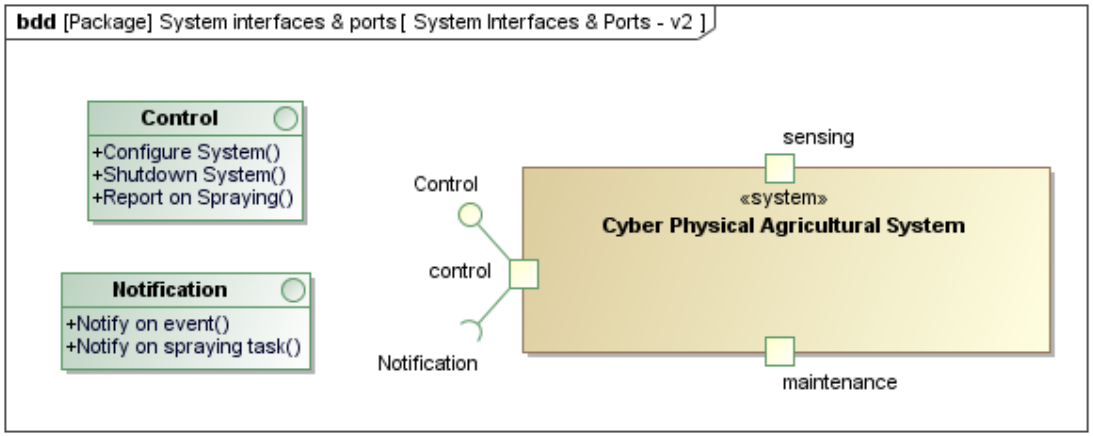
An interface declares a set of public features and obligations that constitute a coherent service offered by a classifier.

- An interface specifies a contract; any instance of a classifier that realizes the interface must fulfill that contract.



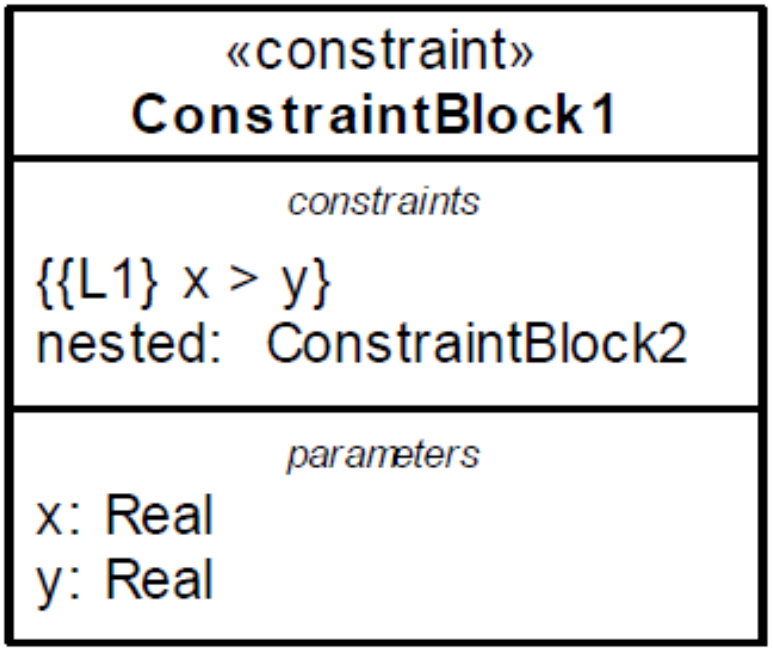
Interface

- Specifies the behavioral features of a block
  - Provided interface
    - A service is provided by the block for its environment
    - Denoted with a lollipop / ball symbol
  - Required interface
    - The set of the operations is required by the block for its operation.
    - Denoted with a socket symbol



Defining Constraint Blocks

- Constraint Block node
  - Specify a network of constraints to
    - Constrain the physical properties of the system
    - Identify critical performance paramaters
  - Constraints represent mathematical expressions
    - F = m*a
    - a = dv / dt



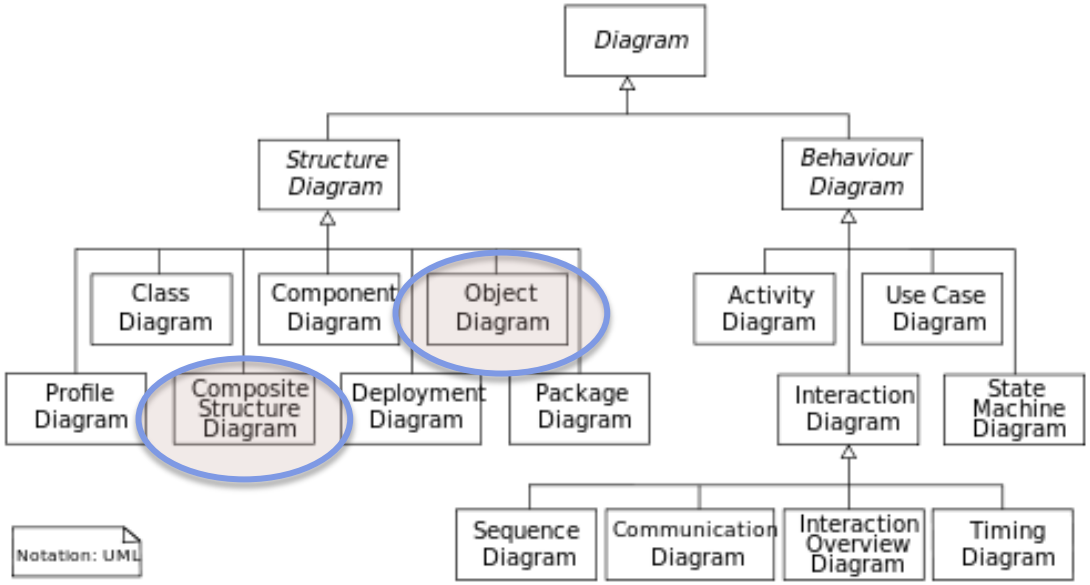
Block Usage

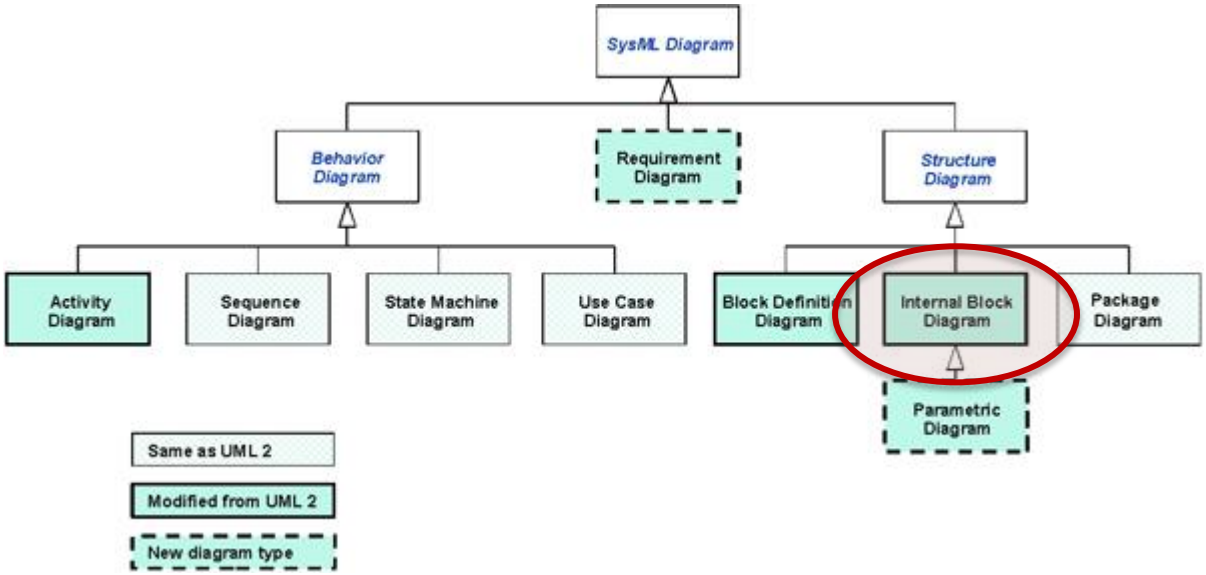
Internal Block Diagrams

What is it about?

Context of the Modeling Aspect

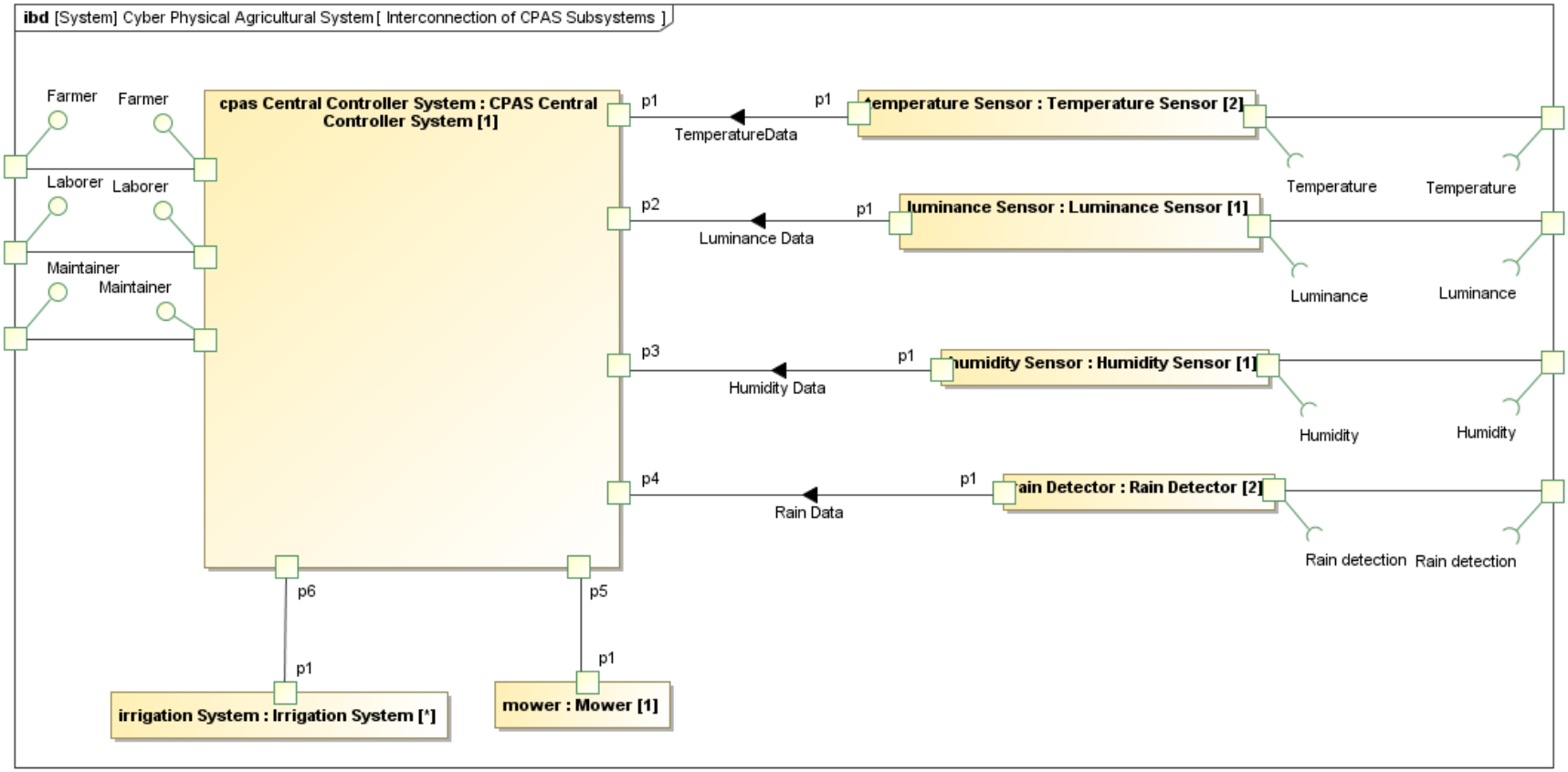
Internal Block Diagram





Modeling Aspect

How are components used in a given context or system?

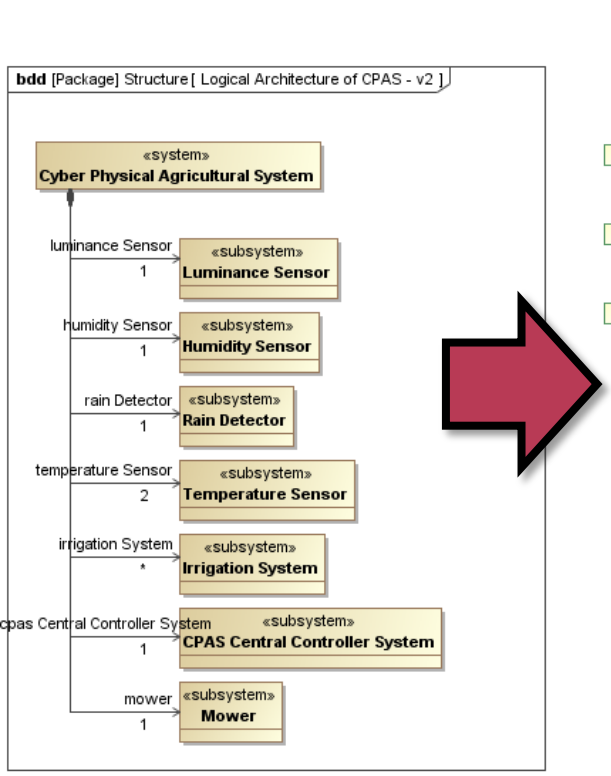


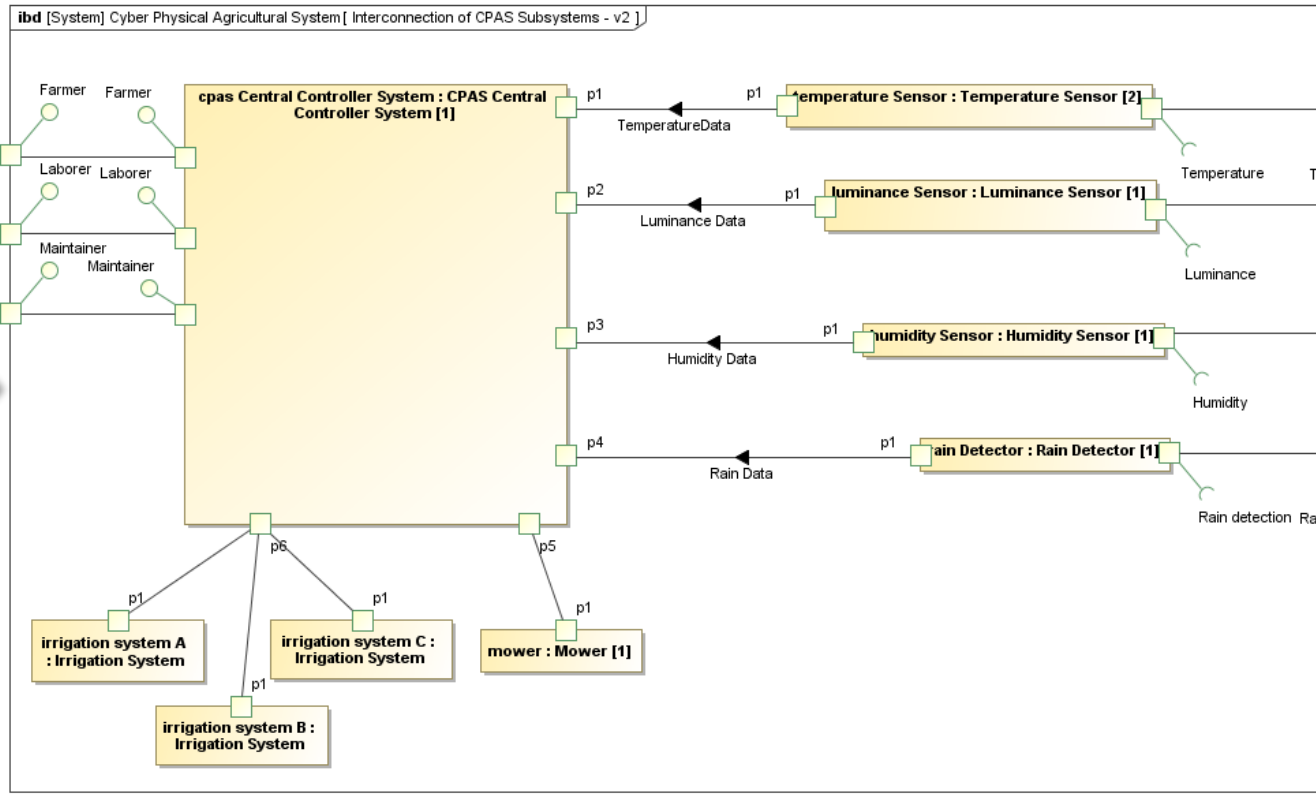
Objectives

- Define how components are interacting with each other within a given system
  - Define relations
  - Define data flow
  - Define interfaces

BDD vs. IBD

- Block Definition vs. Usage
  - Block diagram → Definition of the structure
  - Internal block diagram → Usage of this structure in different contexts





Relations to other aspects

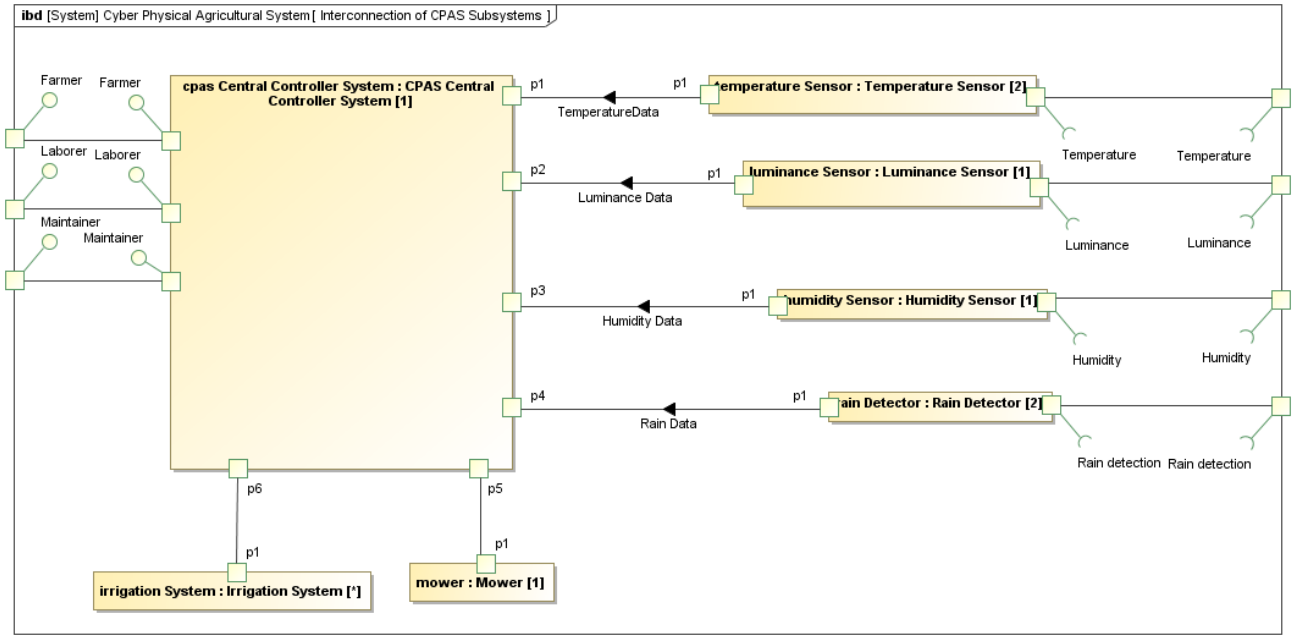
- Interpreted in the context of a block
- Defines usage of blocks
- Item flows can be mapped to object flows in activities

What are the building blocks?

Modeling Elements & Notation

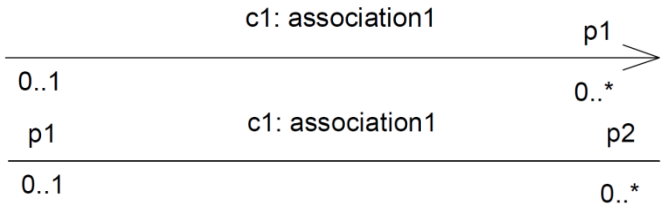
Defining Blocks - 1

- Nodes
  - The instances of the nodes from the related block diagram (Part properties)
  - Can have a unique name with type indication
  - Can also be nested into more levels

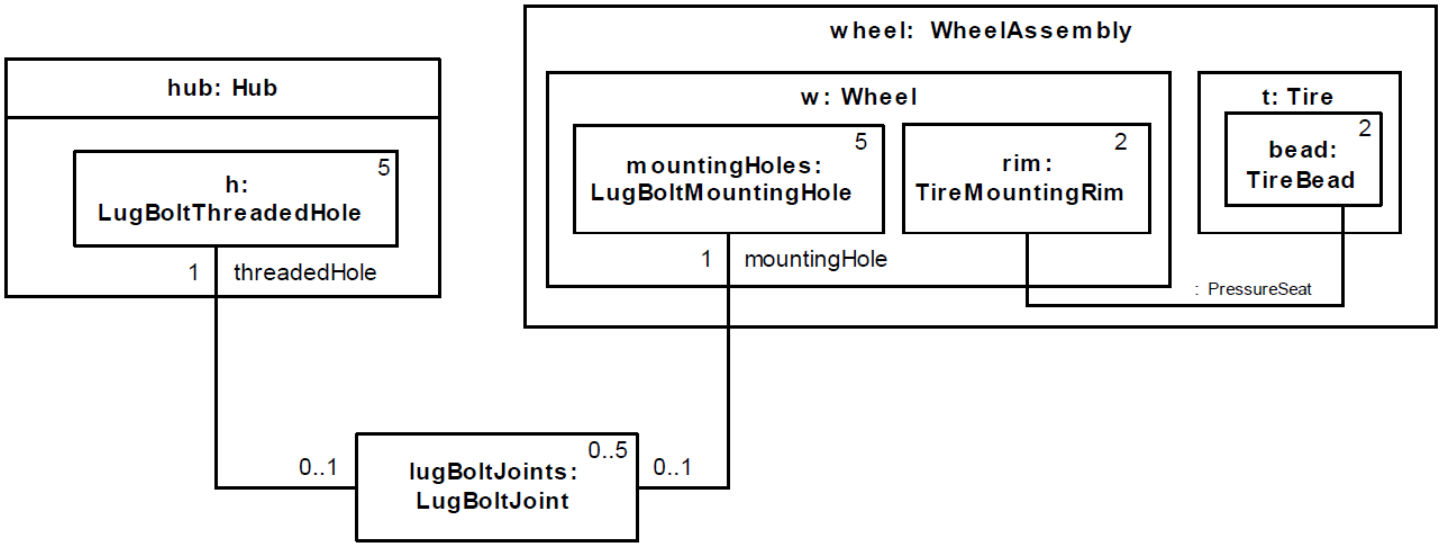


Defining Blocks - 2

- Paths
  - The properties can be connected to each other with
    - Unidirectional Connector
    - Bidirectional Connector

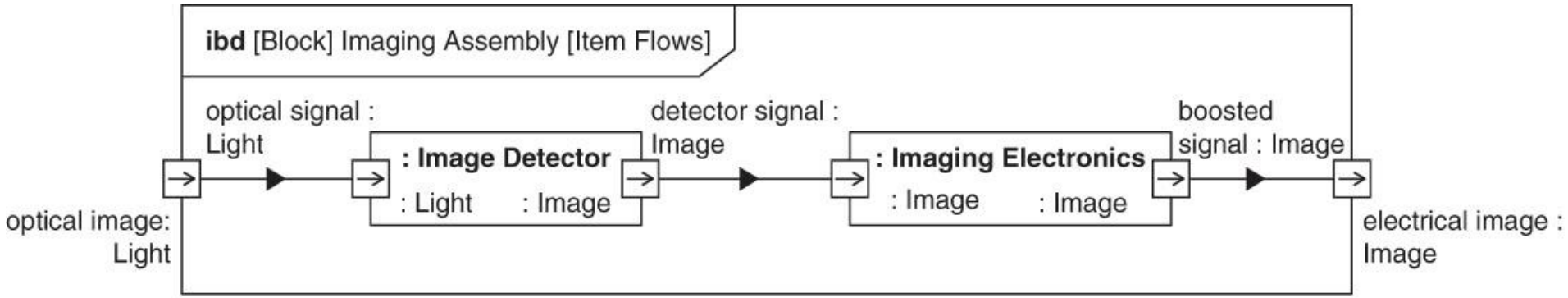


  - Connectors are the instances of the associations
  - Can have multiplicity on both ends



Defining Ports and Flows

- Ports are to define the interaction points of the part property
- Flows are to specify the items that flow across a connector
- Both are understanded in a particular context



Summary

- Block Definition Diagram
  - *What are the elements of the system?*
- Internal Block Diagram
  - *How are elements within a system relates to each other?*