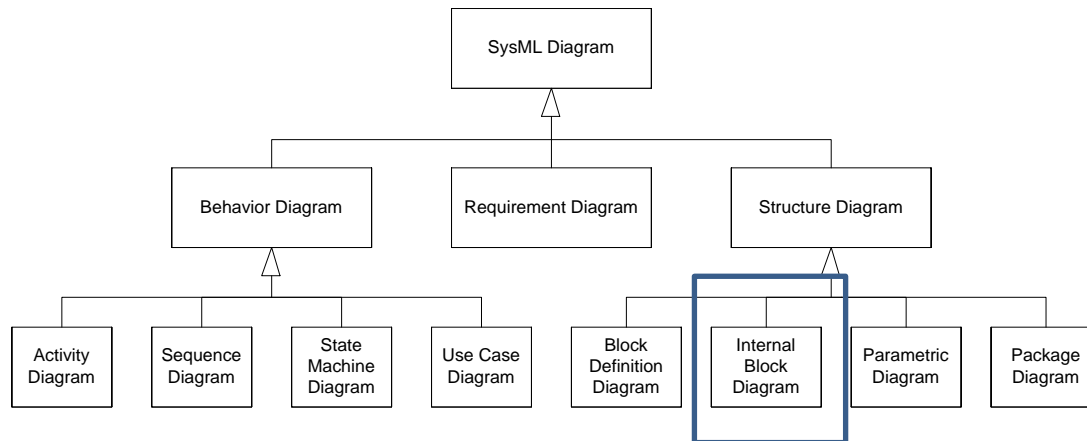


SysML Structural Diagrams 2


Introduction to Systems Engineering
I2ISE

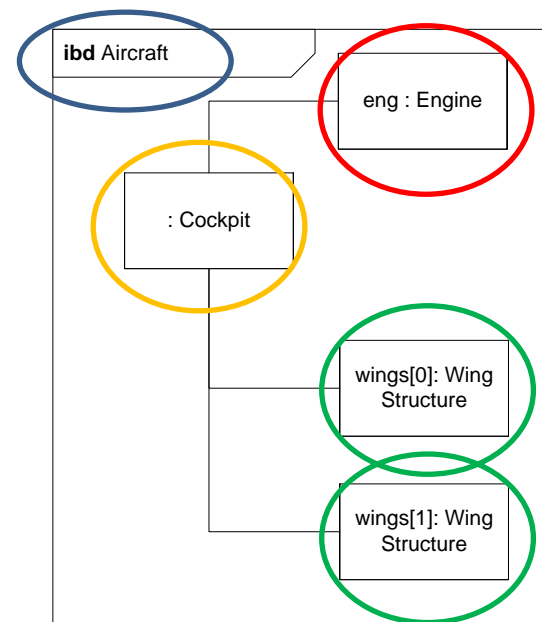
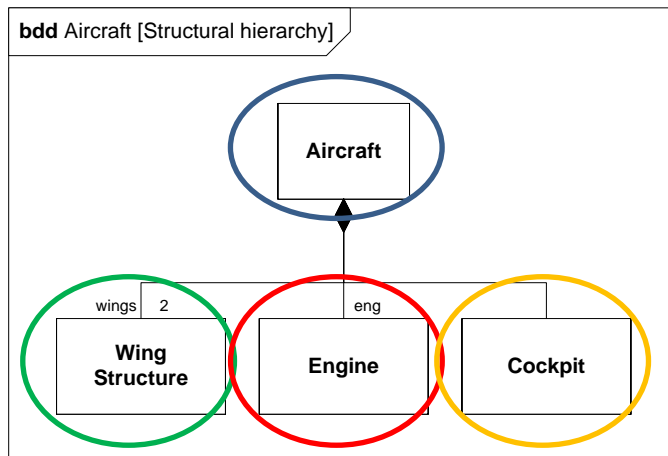
SysML

Internal Block Diagrams



SysML: Internal Block Diagram

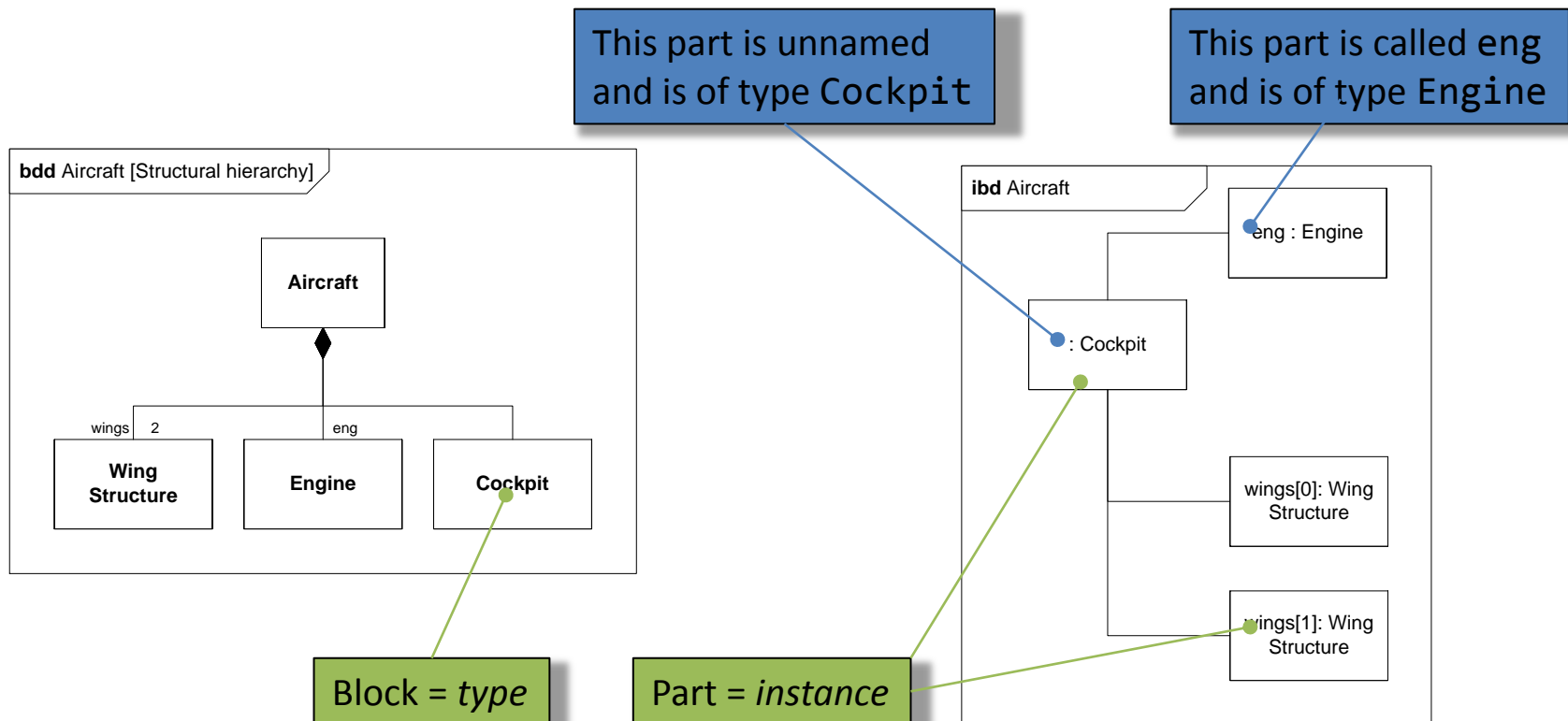
- An *Internal Block Diagram (ibd)* is used to define
 - the *interconnection* and *interfaces* of the parts of a block, and
 - the *information flow* between parts
-  An ibd **always** relates to a block on a bdd. It shows the internal connections of the block's constituents



SysML: Blocks and parts

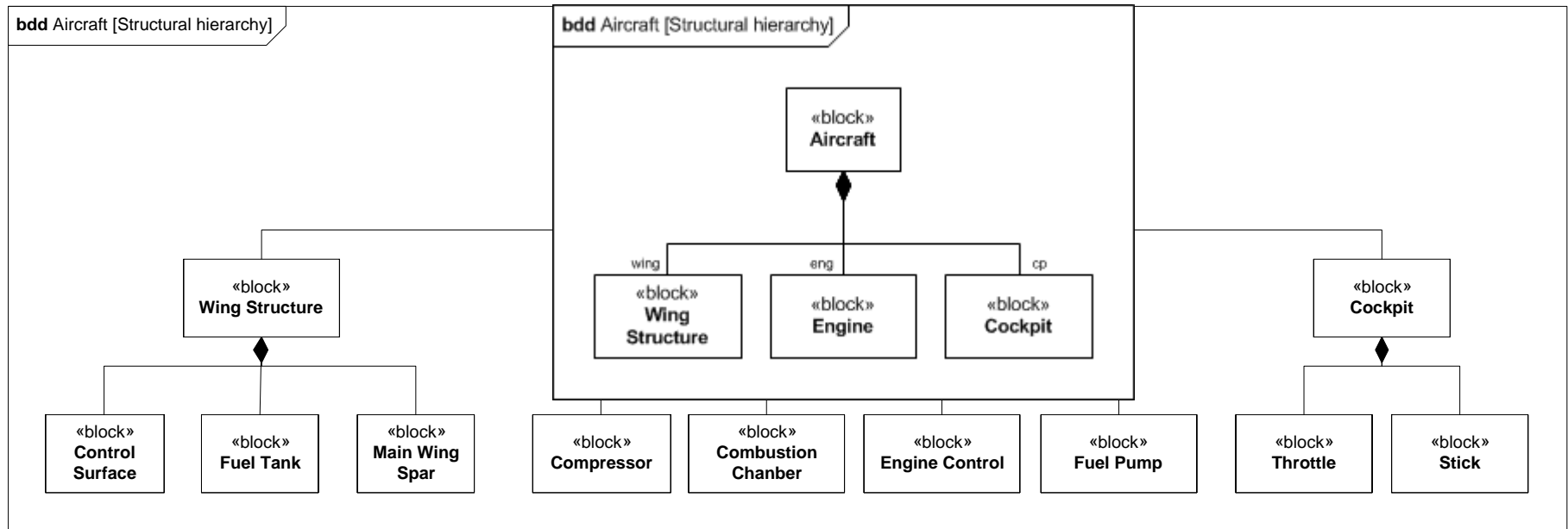


- A block is a *type definition* – there can be only one block with a given name
- A part is an *instance* of a block – there can be many instances of the same block

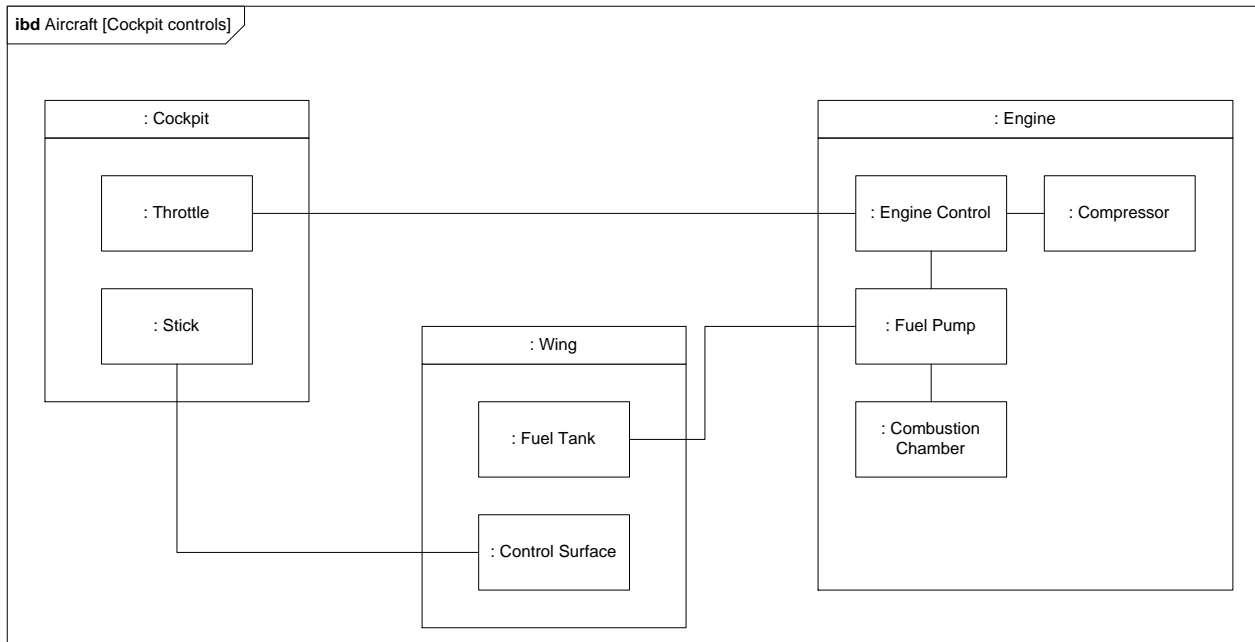
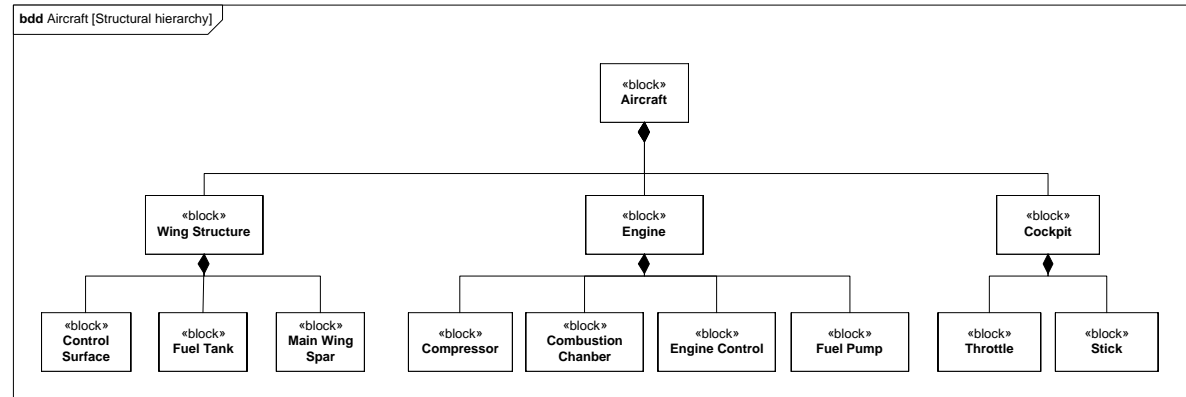
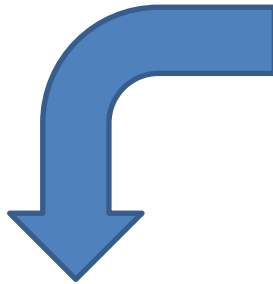


ibd: Aircraft - deep structure

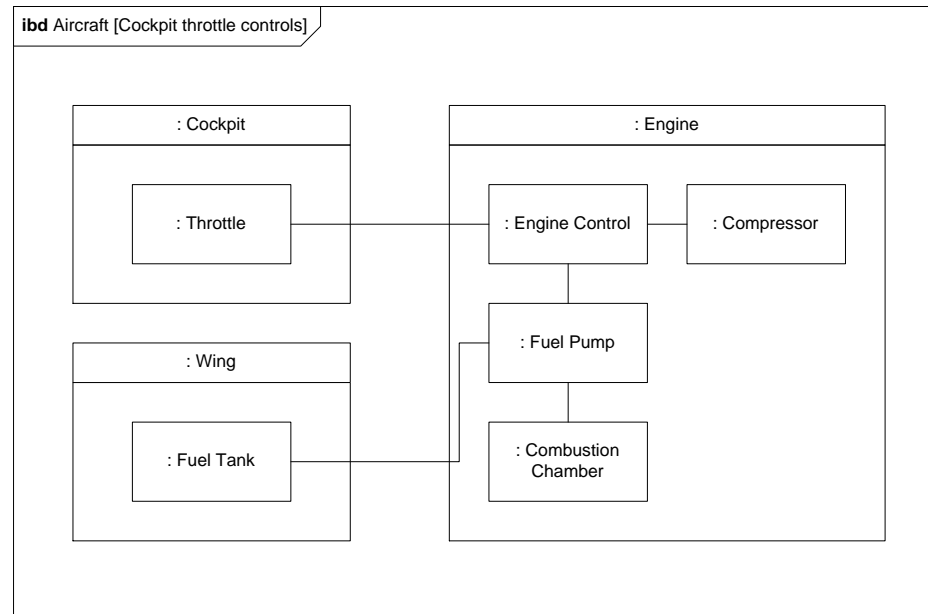
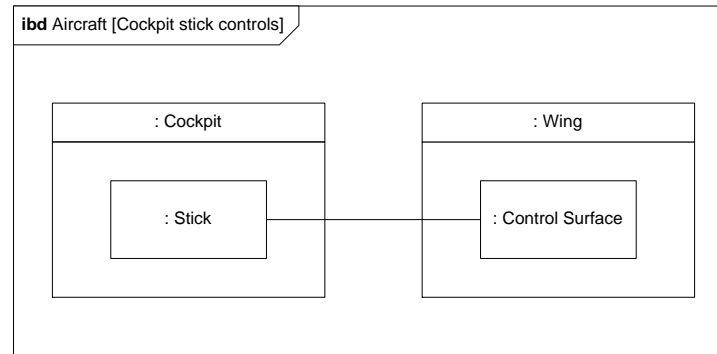
- Deep structure on a bdd can be shown in an ibd:



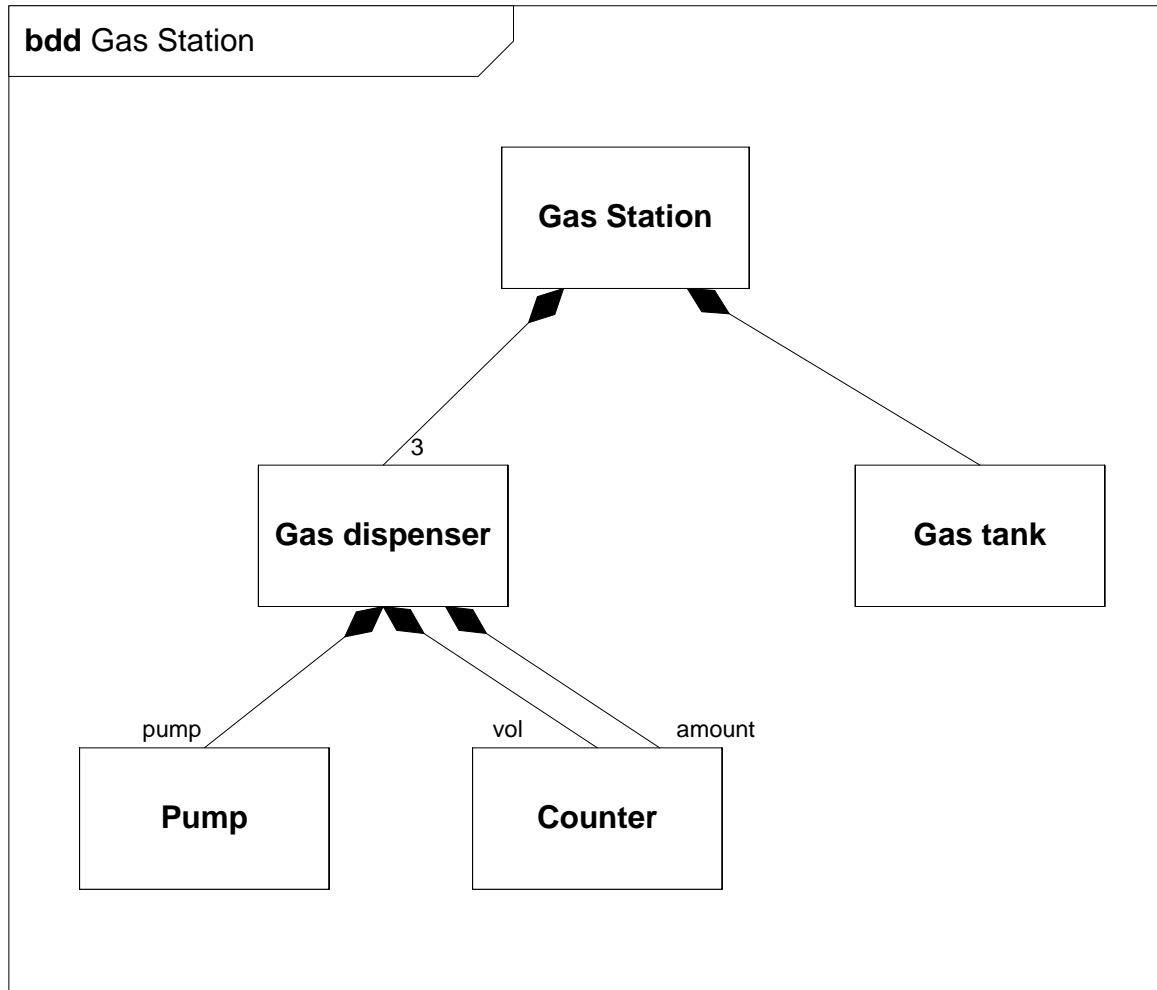
ibd: Aircraft - deep structure



ibd: Aircraft – better deep structure

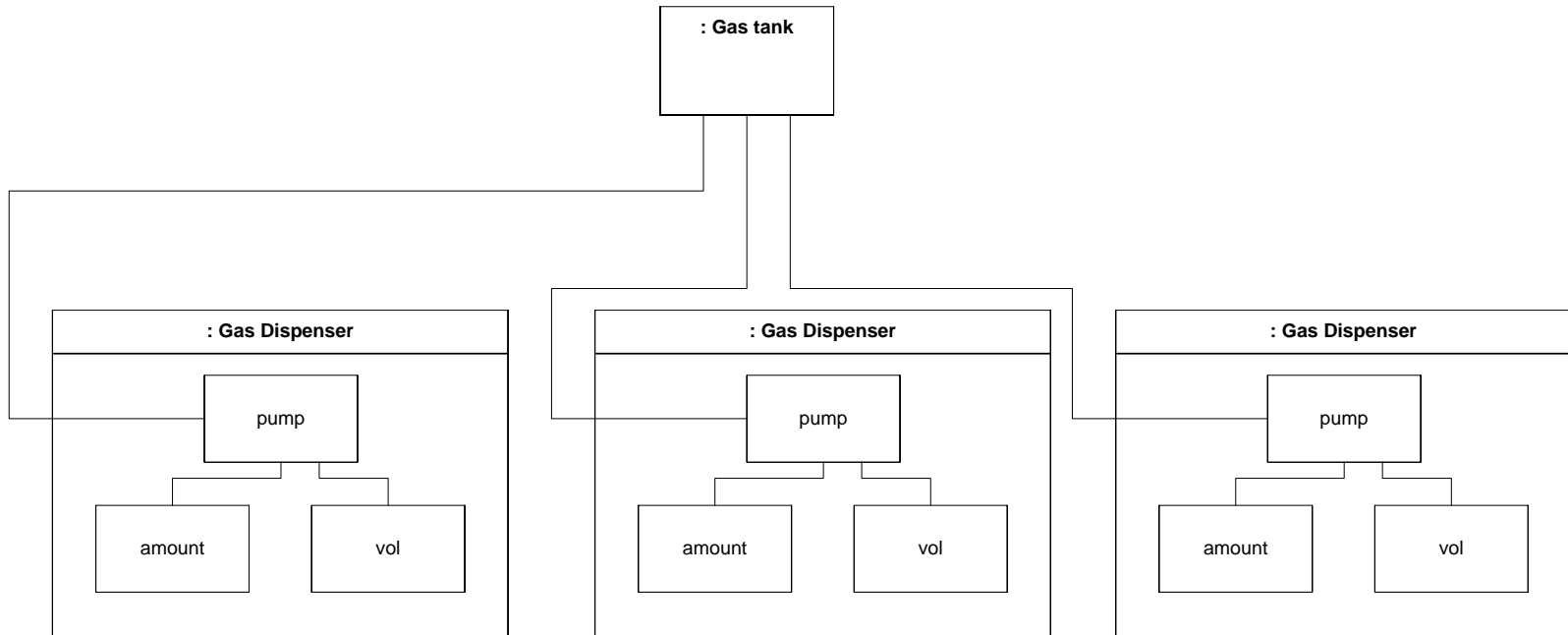


ibd: Gas station example

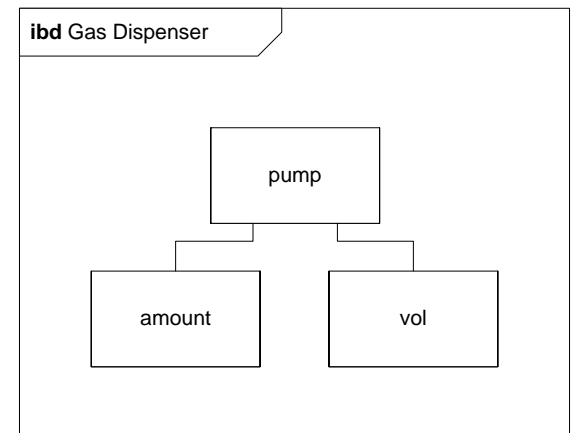
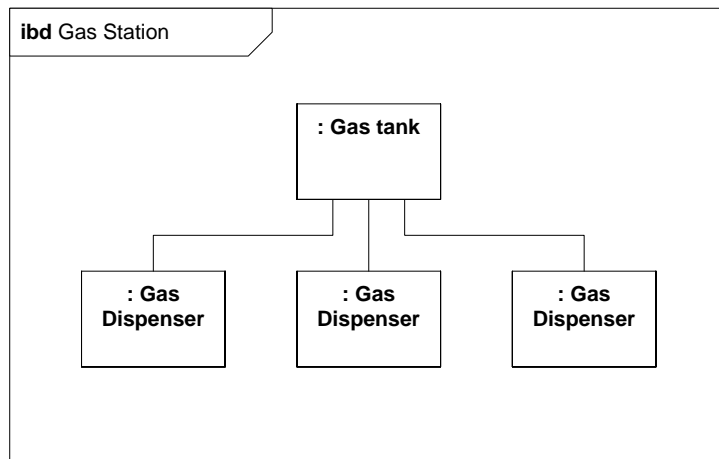


ibd: Gas station example

ibd Gas Station



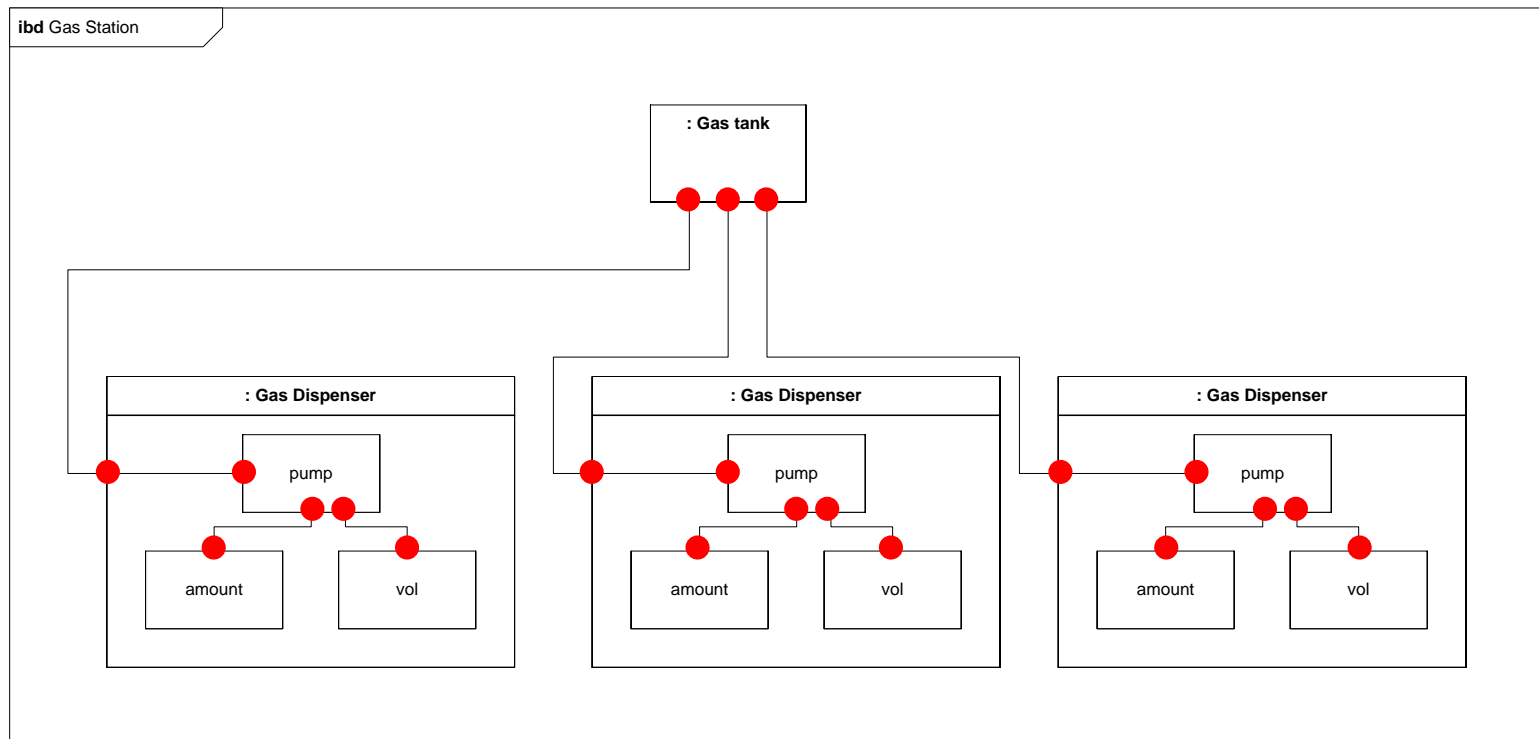
ibd: Better Gas station example





So far, so good...

- We can say a lot about the structure of a system in terms of *blocks* and *parts*...but what about their *interfaces*?



SysML

Modeling interfaces
using *items*, *item flows* and *ports*

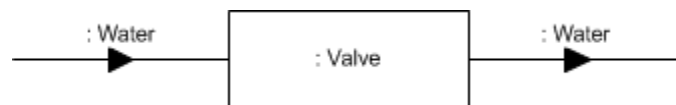
Modeling interfaces

- We would like to express more about the *connection* between parts on the ibd
 - This would help us to define the *interface* of the parts
- To do this, we must define *items*, *item flows* and *ports*!



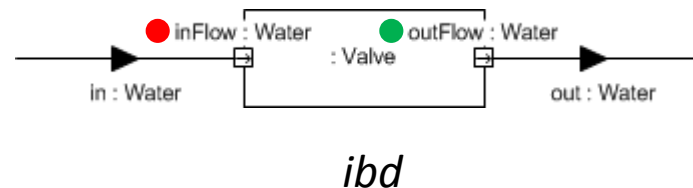
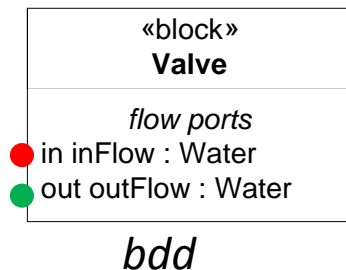
Items and item flows

- An *item* describes an entity that flows through a system (blocks, value types or signals)
 - Physical flow, information flow, energy, ...
 - Simple or complex
- An *item flow* is used to describe a flow of items (!) on a connector between two blocks on an ibd
 - Item flow = item *type* + flow *direction*



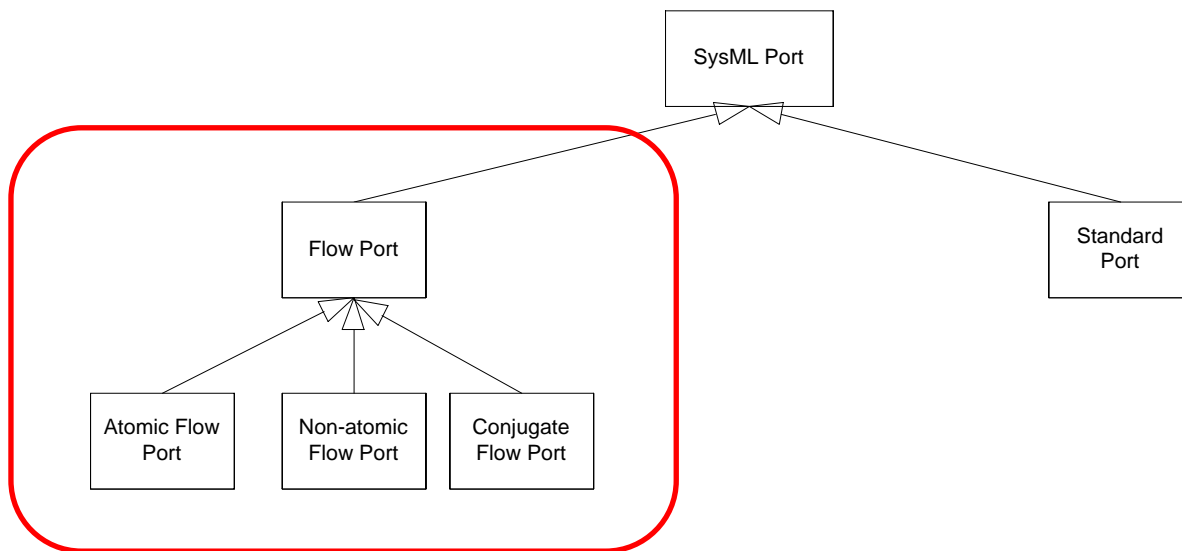
Ports

- A *port* is an interaction point on the boundary of a block
 - Ports are where the items flow into / out of
 - One block can have many ports
- Ports are *defined* on the blocks on a bdd and used to connect *parts* on ibds



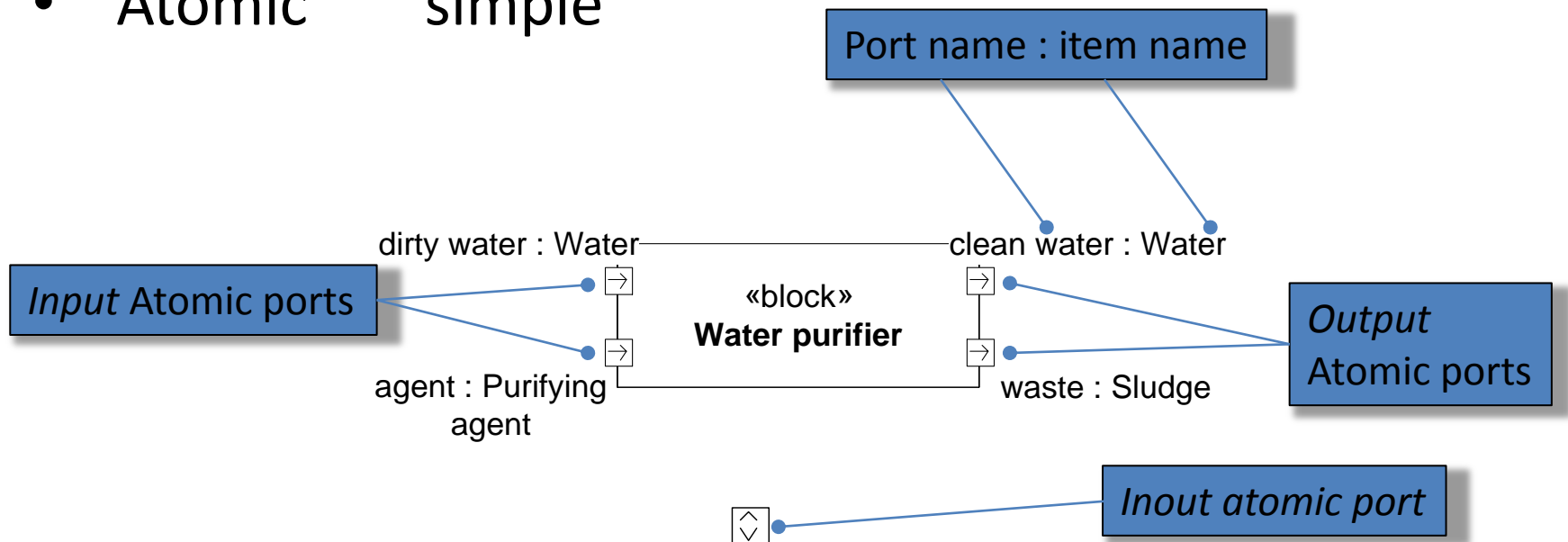
Ports

- Ports come in different flavours, each with different meaning and use
- We will concentrate on *flow* ports



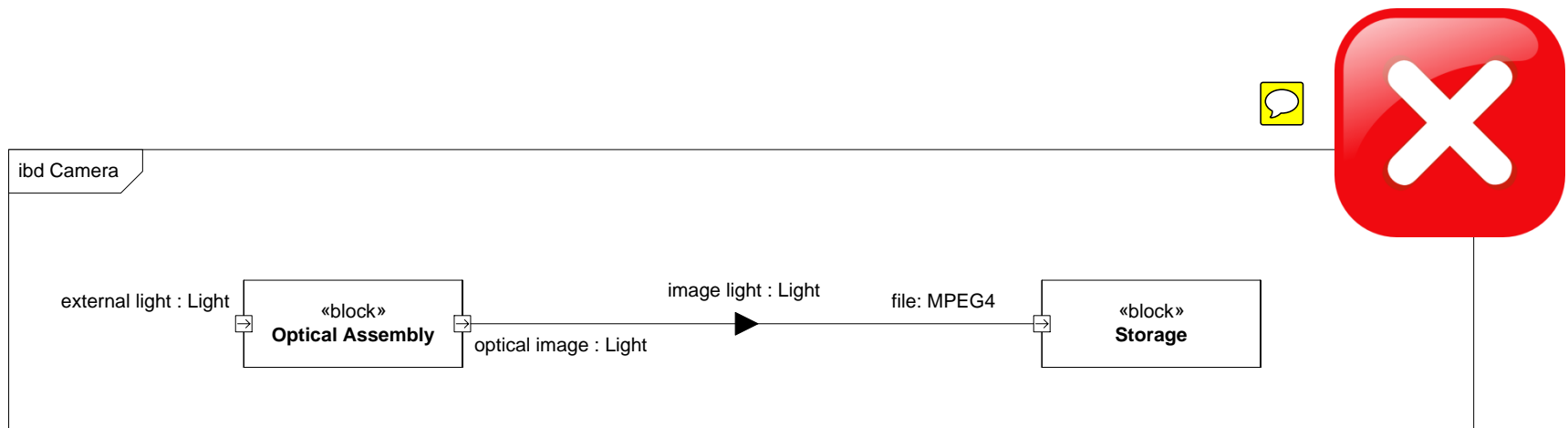
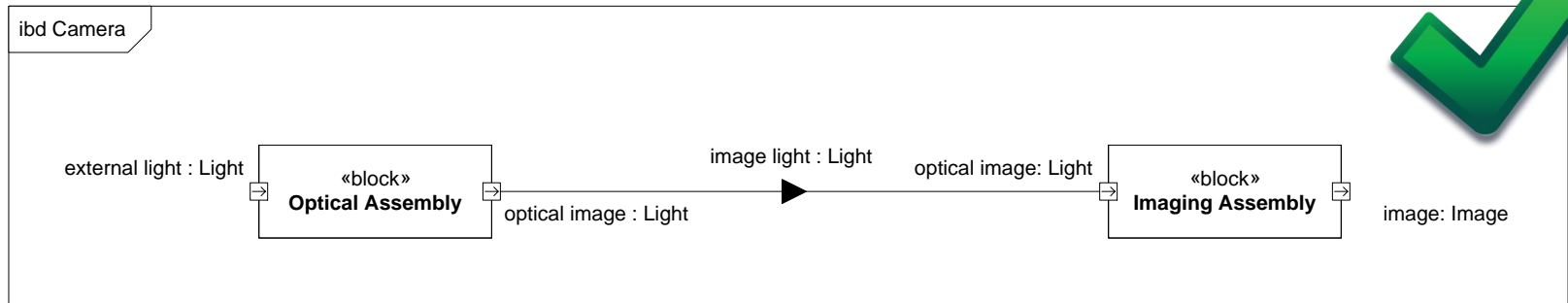
Atomic flow ports

- *Atomic flow ports* are used to describe flows of a single, simple type of item flow to/from a block
 - Directions: In, out or inout
- "Atomic" ~ "simple"




Atomic flow ports

- Atomic flow ports can be connected only if directions and item flow are compatible:





Nonatomic flow ports

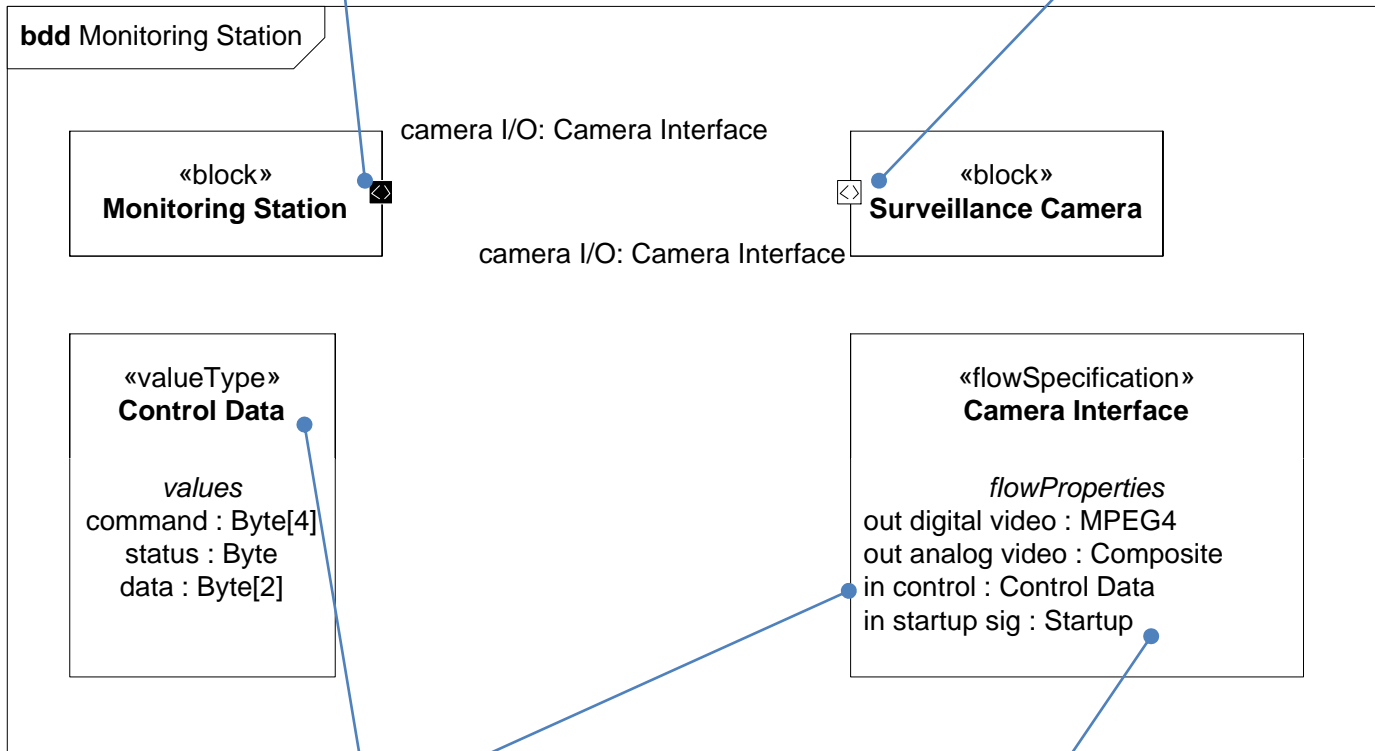
- Nonatomic flow ports are used for composite interfaces
 - “Nonatomic” ~ “composed of several things”

- A nonatomic flow port must be matched by a *flow specification* on a bdd
 - Each component given as a flow property (type and direction)
- You may also use a *conjugate flow port* (see next slide)

Nonatomic flow ports



Conjugate nonatomic flow port

Nonatomic flow port

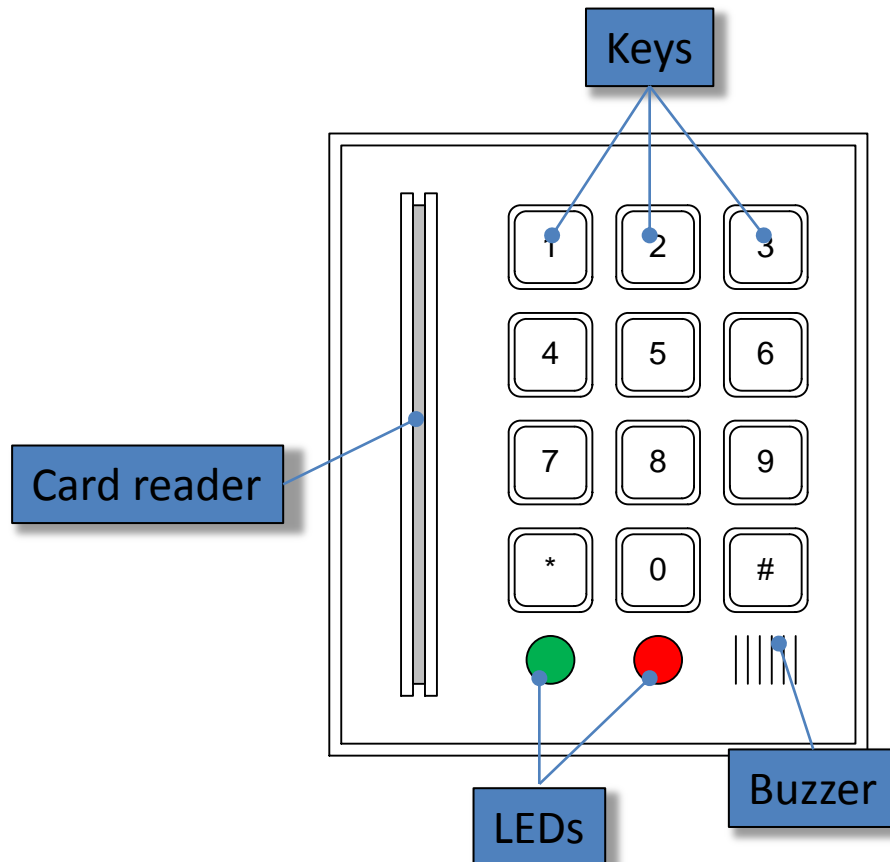


Specification of the Control Data value type used in the flow specification

Flow specification for port

Your turn!

- Given a bdd for an access control system, create ibd incl. ports and item flows



Your turn!

