# Introduction to SysML

# Part 7.0: Internal Block Diagrams

With tutorial exercises using Magicdraw

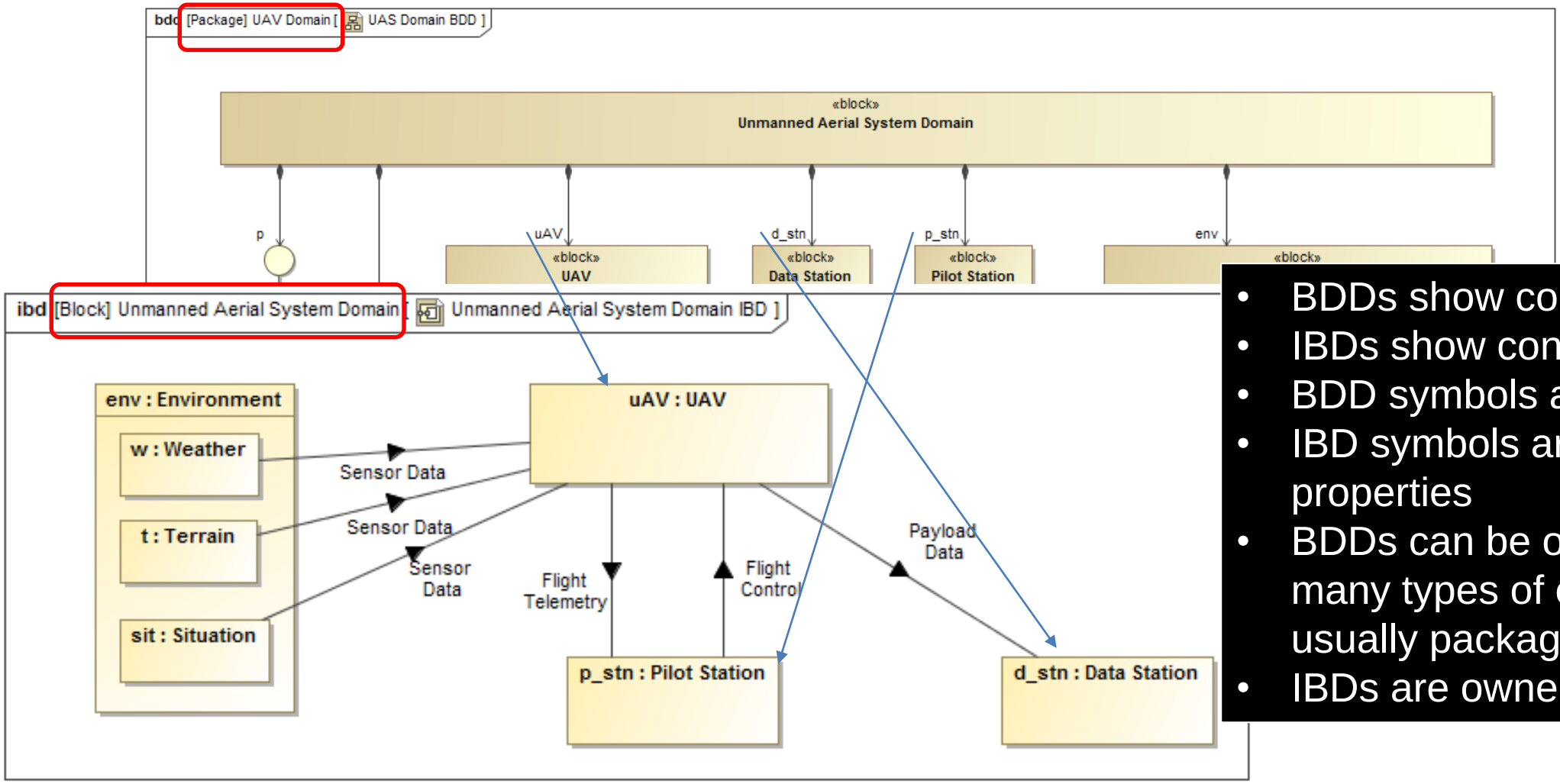
# Learning Objectives

- Part Properties, Connectors and Item Flows
- Internal Block Diagrams
- First Exercise, creating a simple internal block diagram
- Proxy Ports
- Second Exercise, creating an IBD with proxy ports
- Allocations
- Association Blocks, Full Ports and Item Properties
- Third exercise, creating multilevel ports

# Internal Block Diagrams

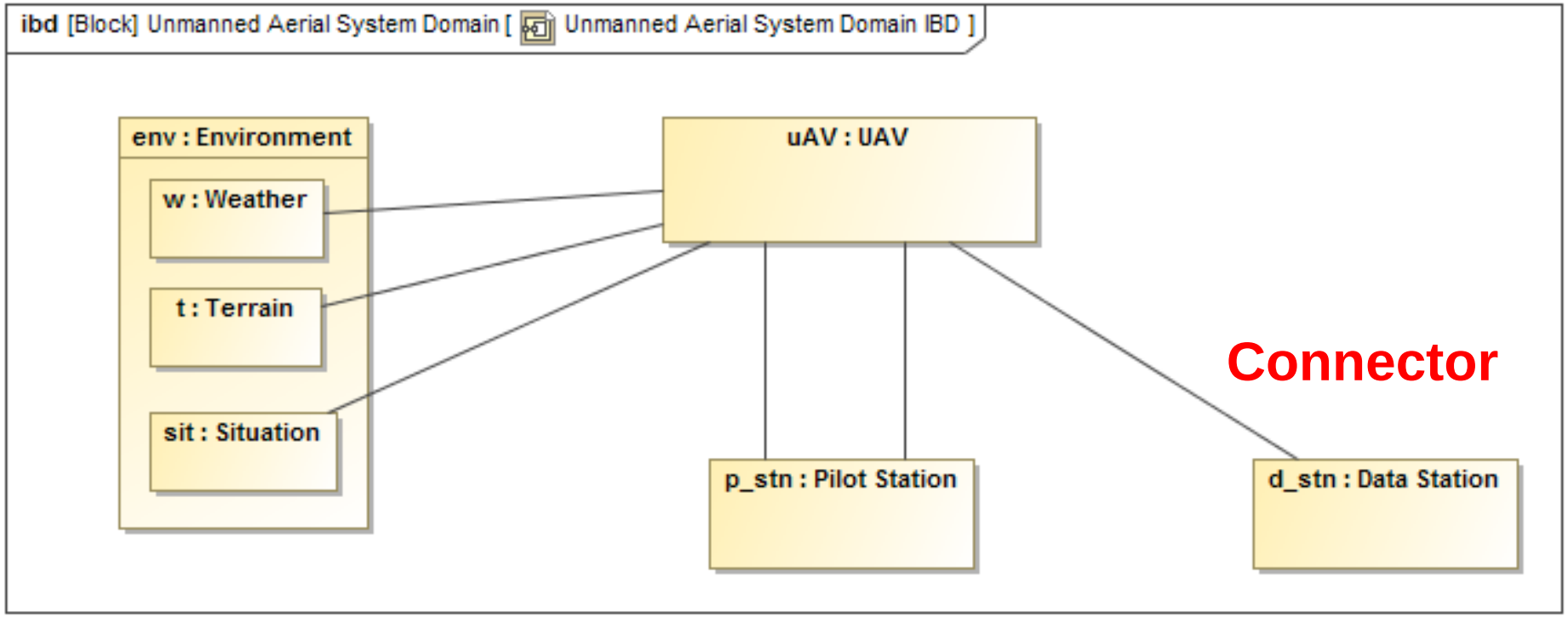
- Internal Block Diagrams (IBDs) are the third kind of SysML structure diagram
- An internal block diagram is a diagram that shows the internal structure of a block
- A Parametric Diagram is owned by a Block
- It shows the value properties of that block and how they are connected
- An Internal Block Diagram is owned by a Block
- It shows the part properties of that block and how they are connected
- PARs are useful for mathematical analysis
- IBDs are useful for specifying interfaces between parts
- IBDs have developed a rich and continually evolving vocabulary for that purpose

# BDDs and IBDs




- BDDs show composition
- IBDs show connectivity
- BDD symbols are blocks
- IBD symbols are part properties
- BDDs can be owned by many types of elements, usually packages
- IBDs are owned by blocks

# Internal Block Diagrams

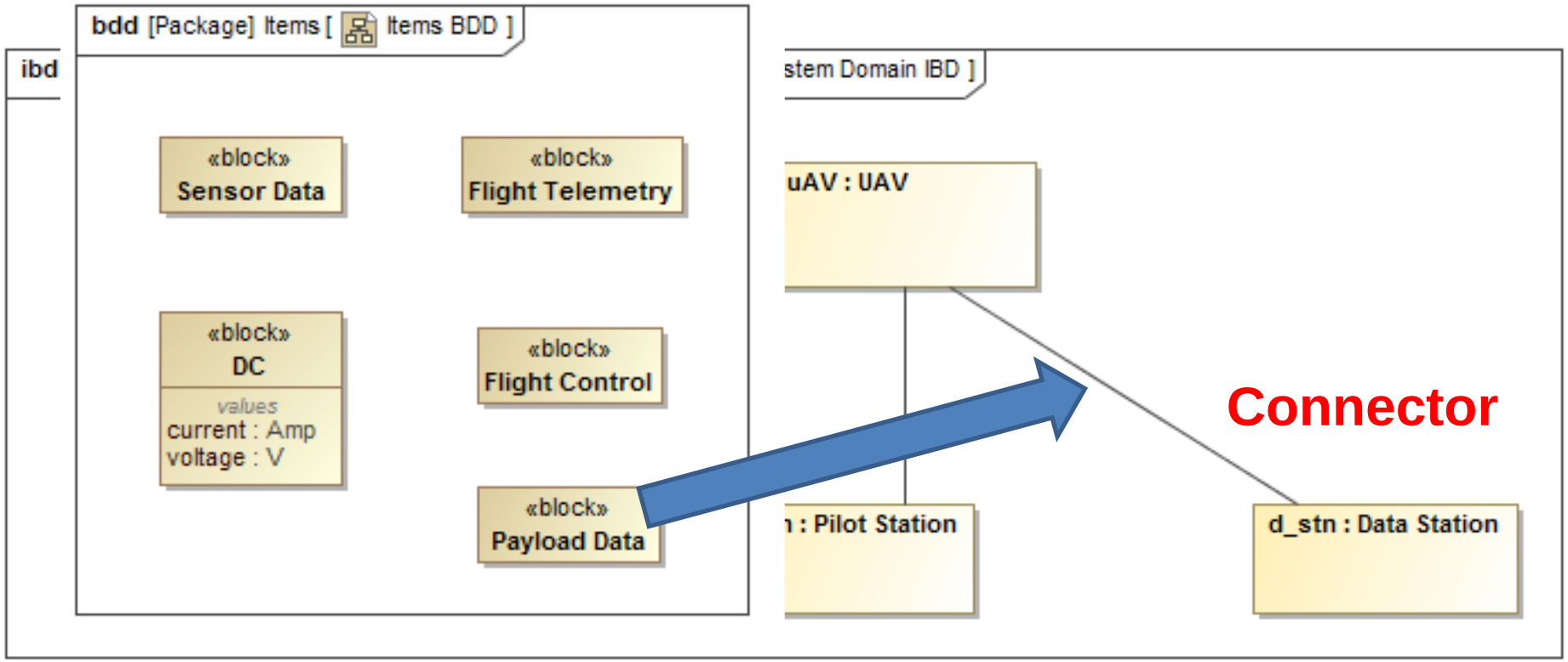



**Connectors represent some kind of connection or interaction between the parts**

**Item Flows represent what passes across that connection in the context of this system**

**Items are blocks, value types or signal, things that might flow in our system**

# Internal Block Diagrams

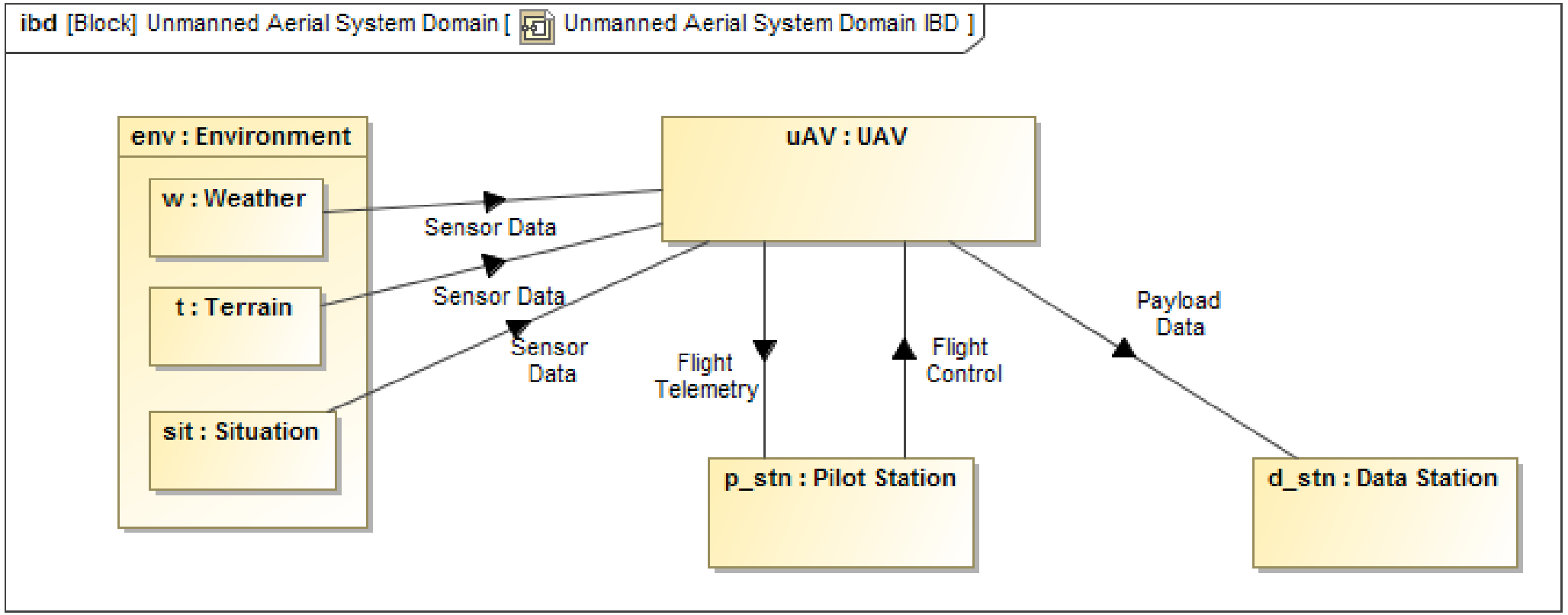


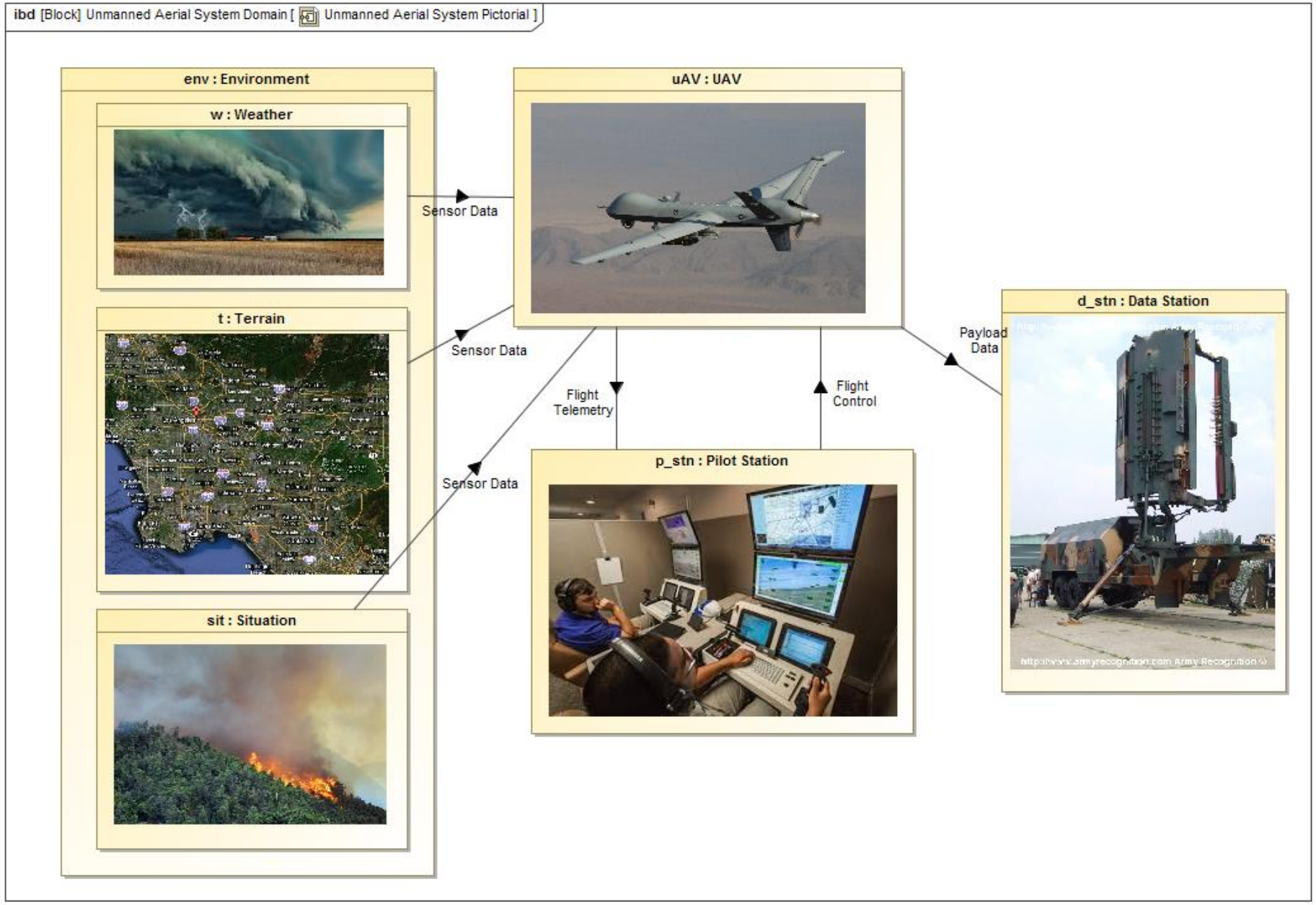

**Connectors represent some kind of connection or interaction between the parts**

**Item Flows represent what passes across that connection in the context of this system**

**Items are blocks, value types or signal, things that might flow in our system**

# First Exercise

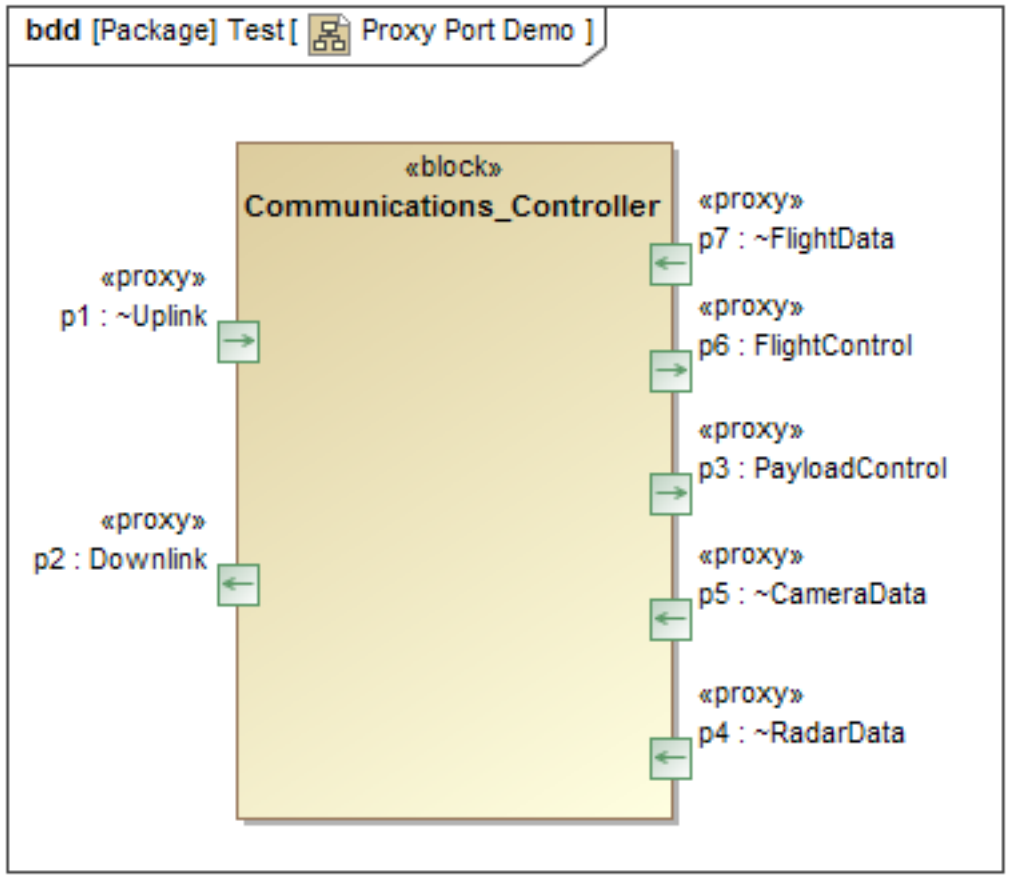


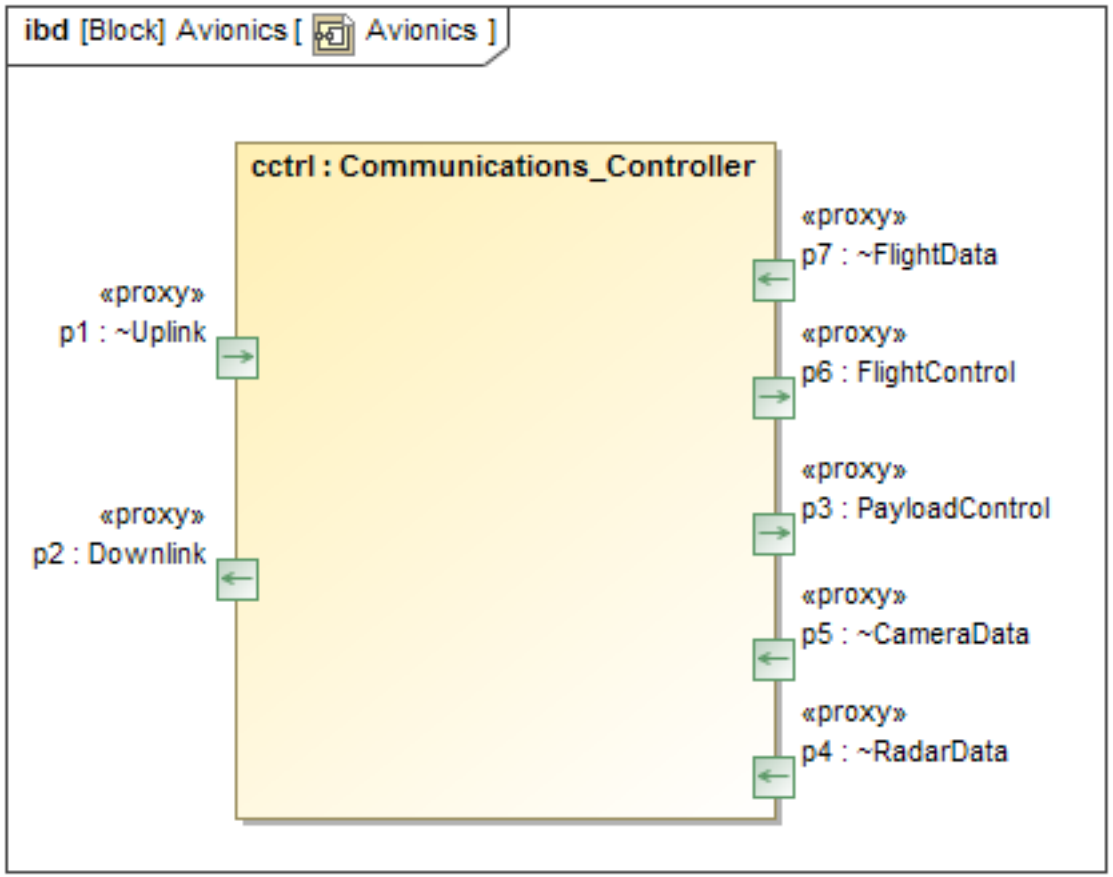


# Proxy Ports

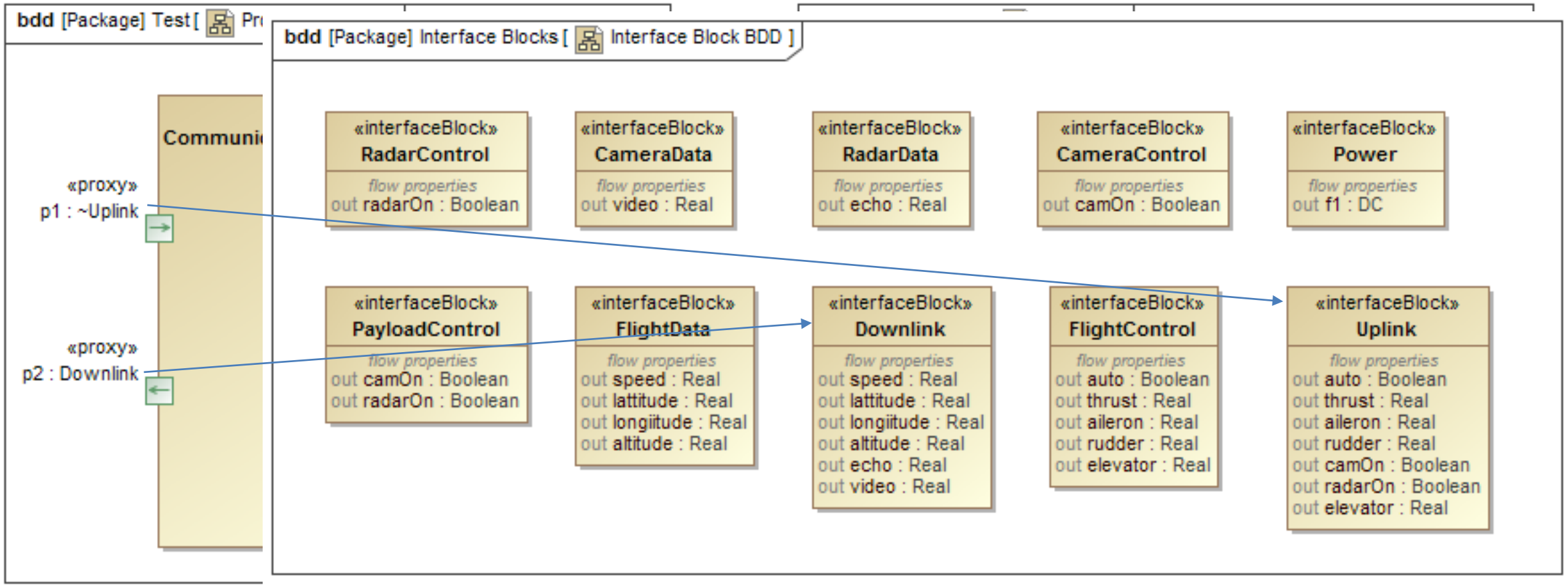
- A Proxy Port is an access point on the boundary of a block that describes/controls what items can flow in and/or out of that block
- Proxy Ports are defined at the Block level, but can appear and be used on Part Properties typed by that Block
- A Proxy Port is typed by an Interface Block
  - An Interface Block contains one or more Flow Properties
    - A Flow Property is typed by an Item and a Direction
- A Proxy Port does not represent a physical component of the system; it is only a functional description in terms of flows (see Full Port)

# Proxy Ports



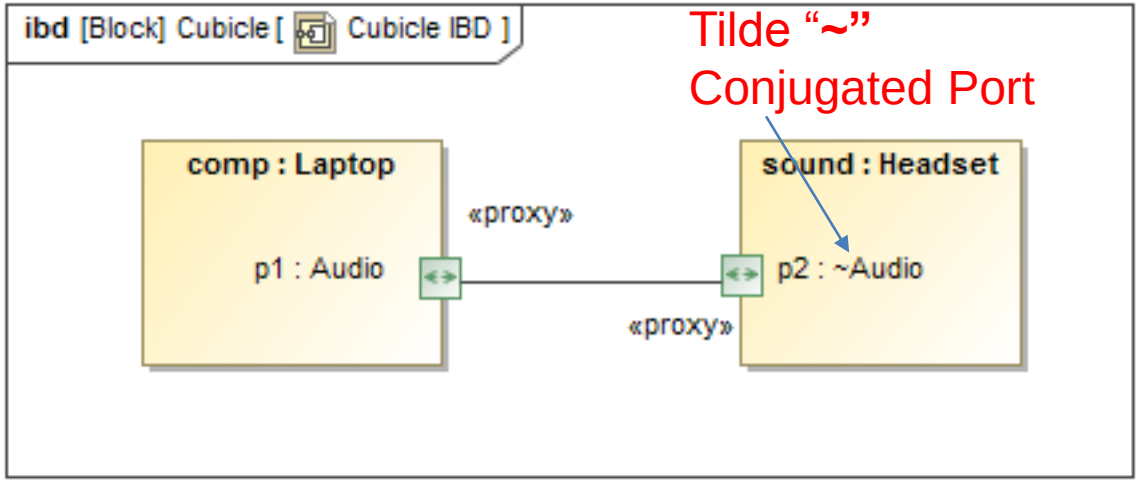



# Proxy Ports

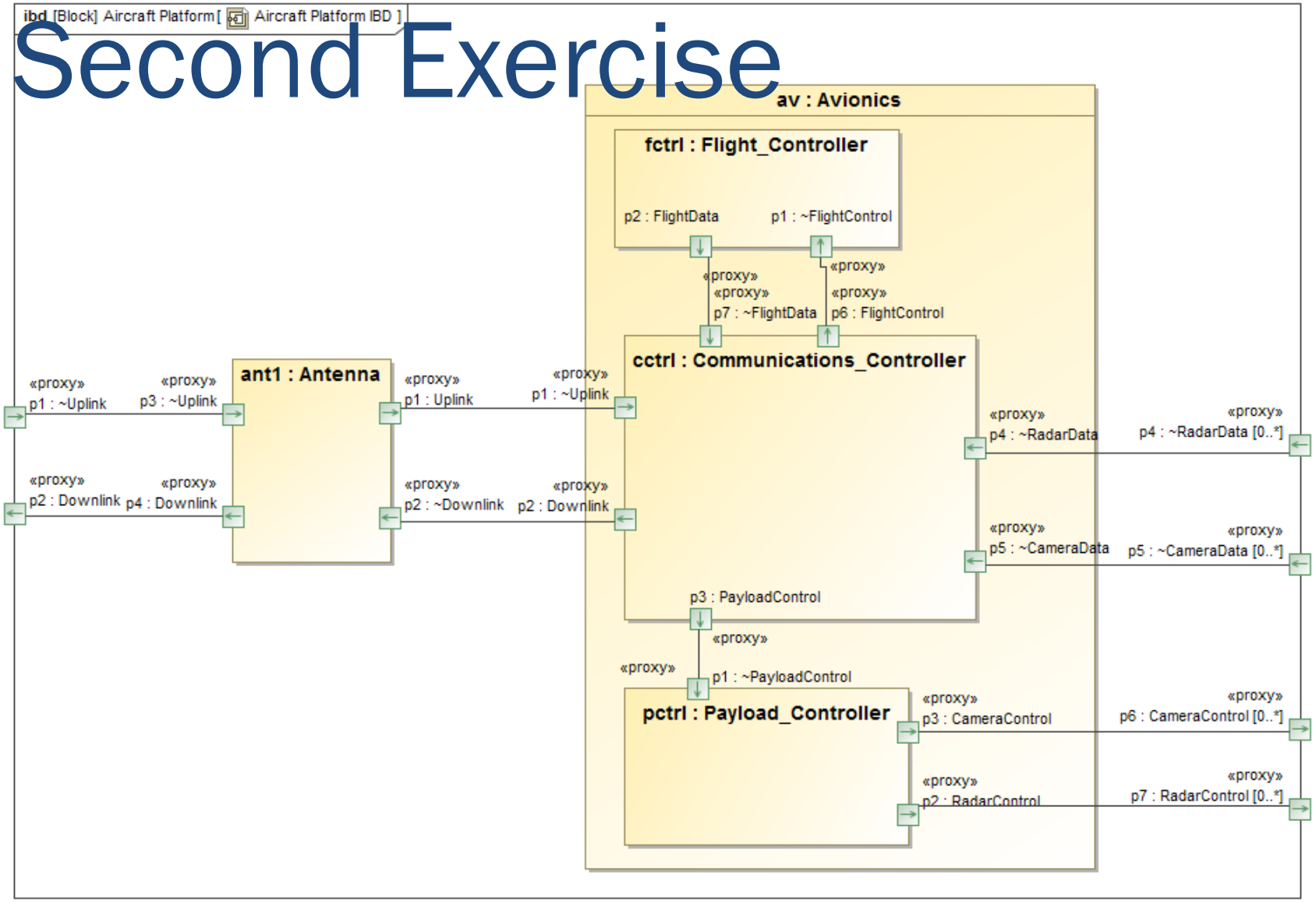


# Proxy Ports

- Proxy Ports are optional in making connections in an IBD
- But they make it easier to validate connections
- Two Proxy Ports that are connected must be compatible in terms of Flow Properties, both Items and Directions
- Validation is a function of the tool, not the language



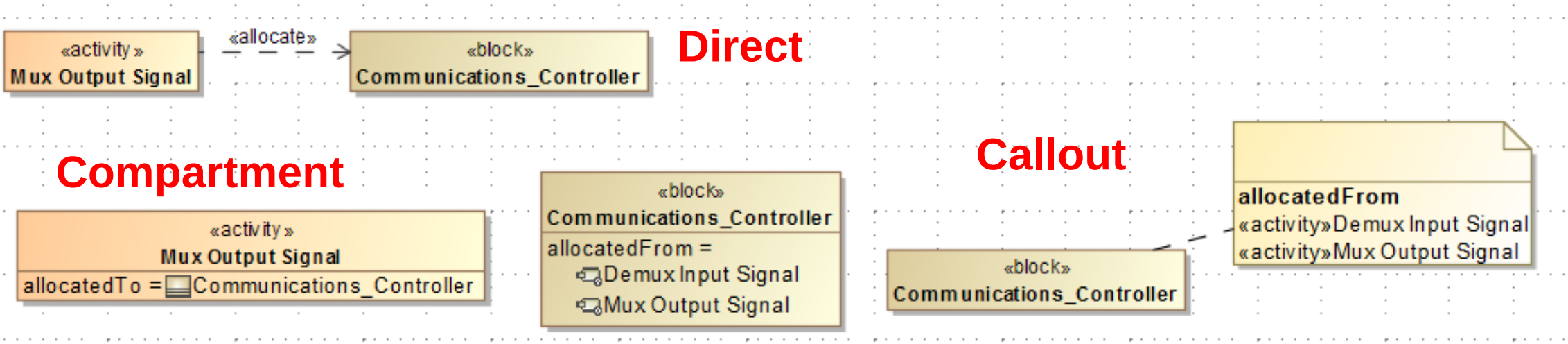

# Second Exercise



# Allocation

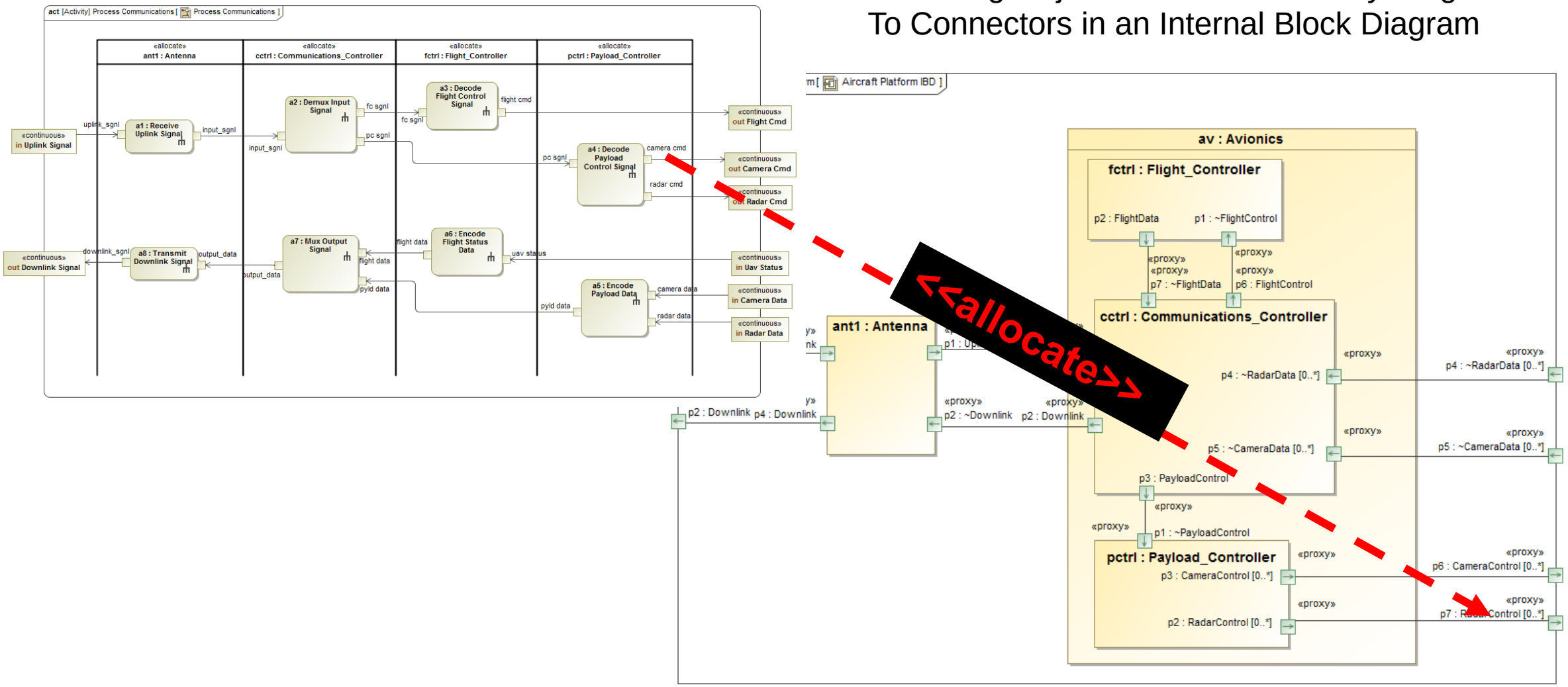
- Allocation is a general-purpose Relationship supported by SysML modeling tools
- You assign a meaning to it, based on the needs of your methodology
- Like other Dependencies, it can appear in a variety of formats in diagrams

# Allocation



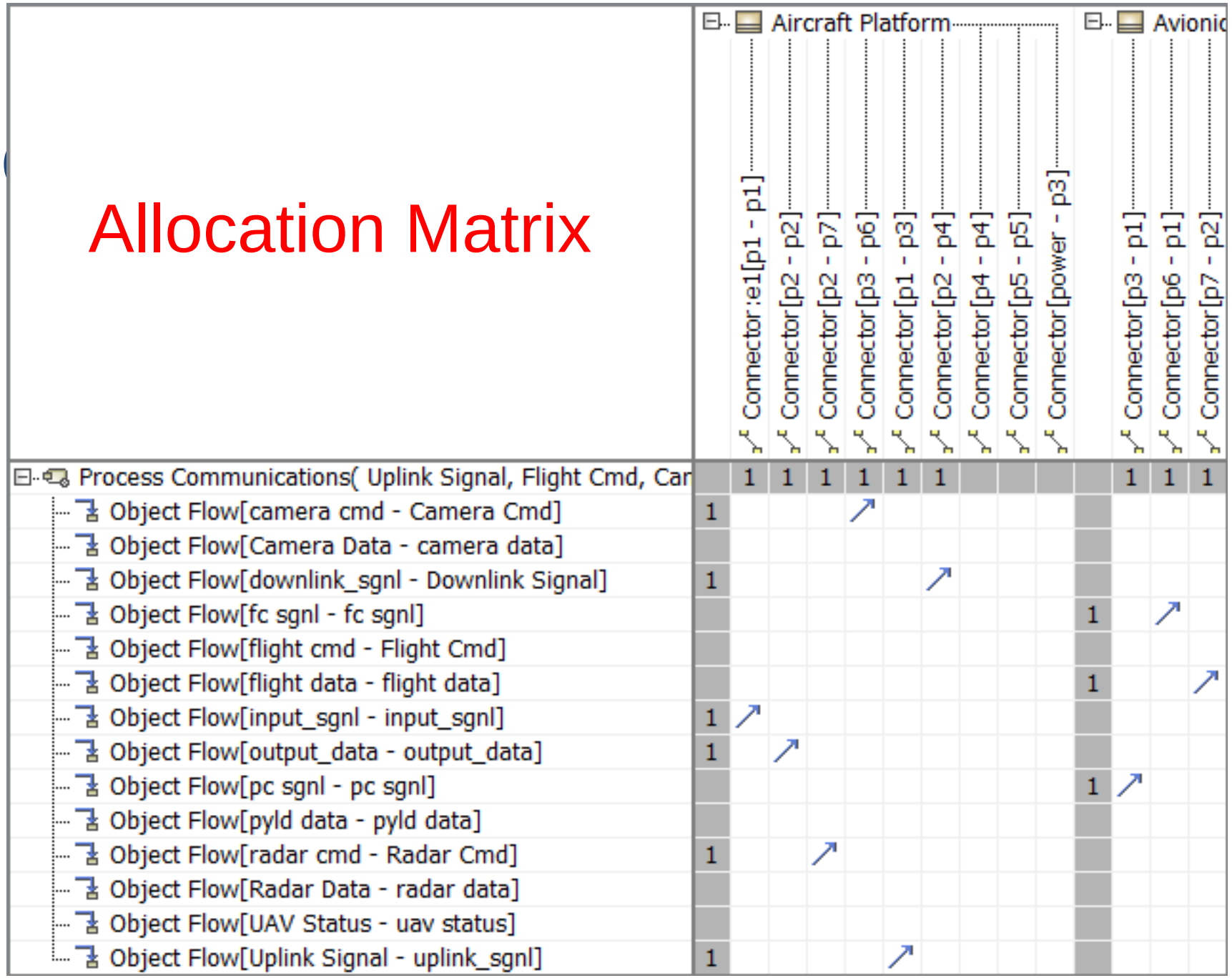
# Allocation

Allocating Object Flows in an Activity Diagram
To Connectors in an Internal Block Diagram



# All

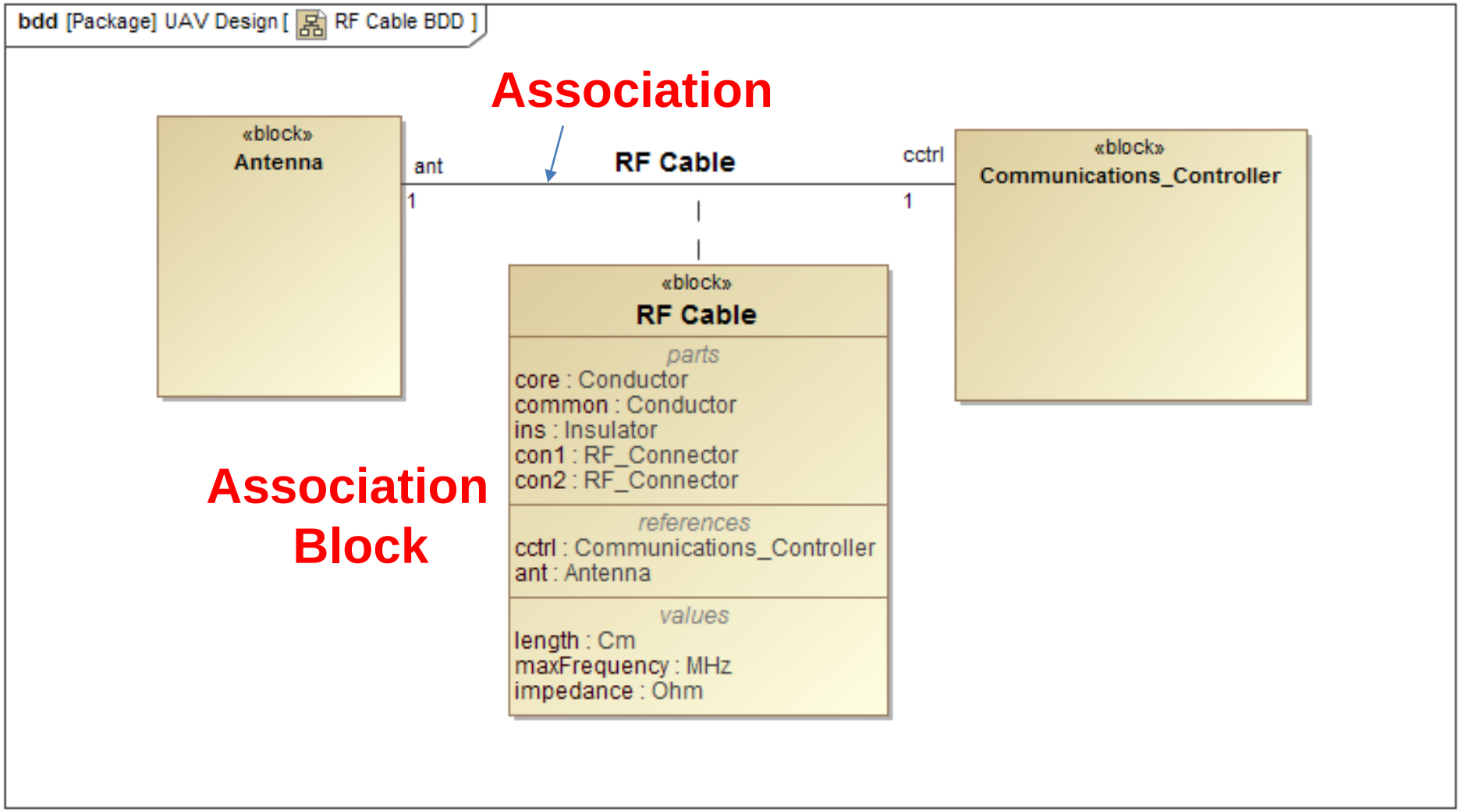
## Allocation Matrix



# Association Block

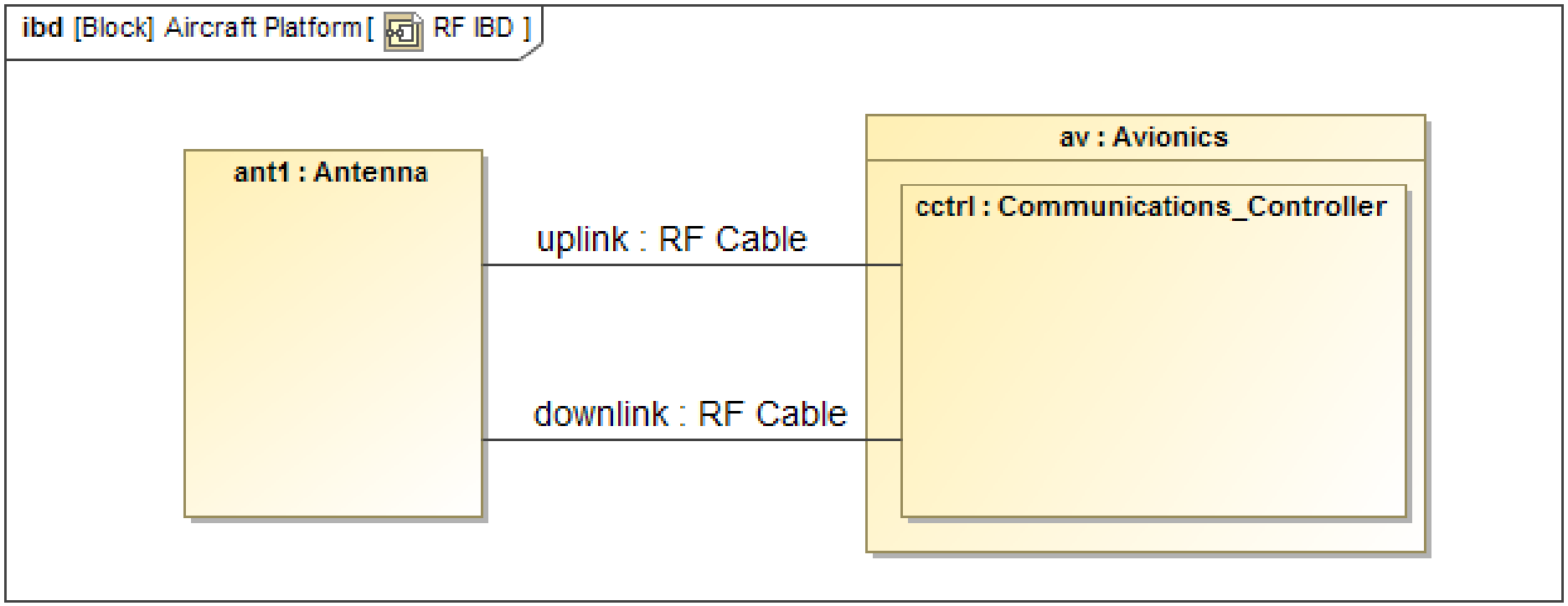
- Connectors can represent actual subsystems, e.g. cables
- SysML uses Association Blocks to contain the subsystem description
- These are defined in BDDS, used in IBDs

# Association Block



# Association Block

- Connectors can represent actual subsystems, e.g. cables
-
-



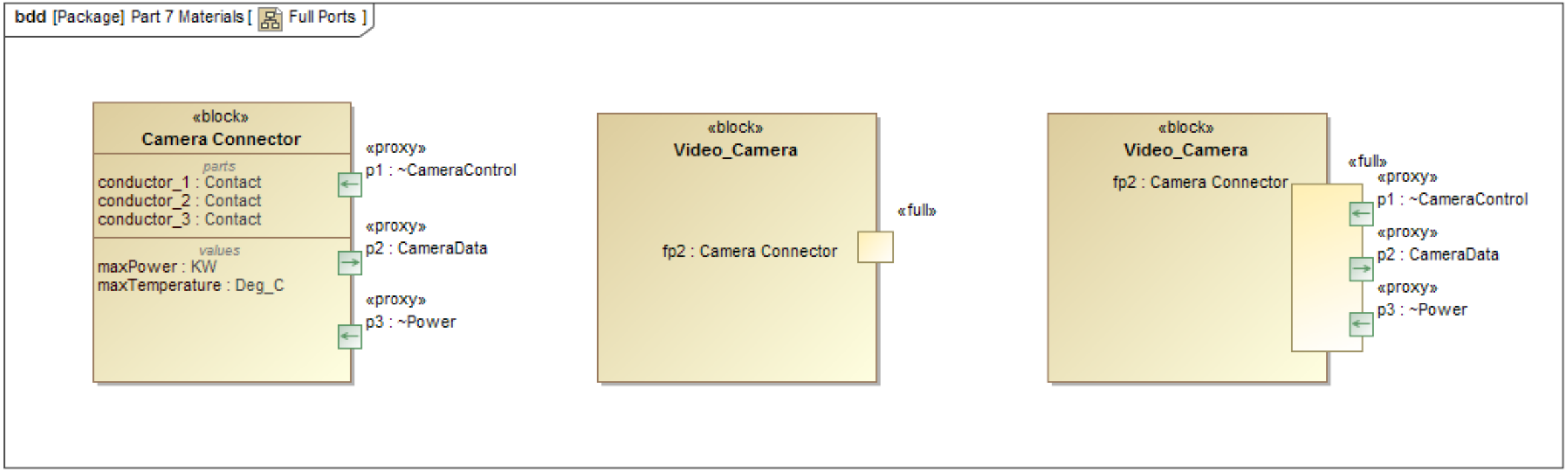
# Full Ports

- A Full Port is an access point on the boundary of a block that describes/controls what items can flow in and/or out of that block
- And describes the physical characteristics of the component that provides that access, a Part and a Port combined
- A Proxy Port is typed by an Interface Block
    - An Interface Block contains one or more Flow Properties
        - A Flow Property is typed by an Item and a Direction
- A Full Port is typed by a Block (like a Part Property)
    - A Block can have Flow Properties, Part Properties, Value Properties…
- There are other types of ports (Flow Ports, Standard Ports) that play a smaller role in SysML

# Full Ports

- Full Ports are typed by Blocks, which can have Ports of their own.
- Modelers can use that to build multilevel interfaces.

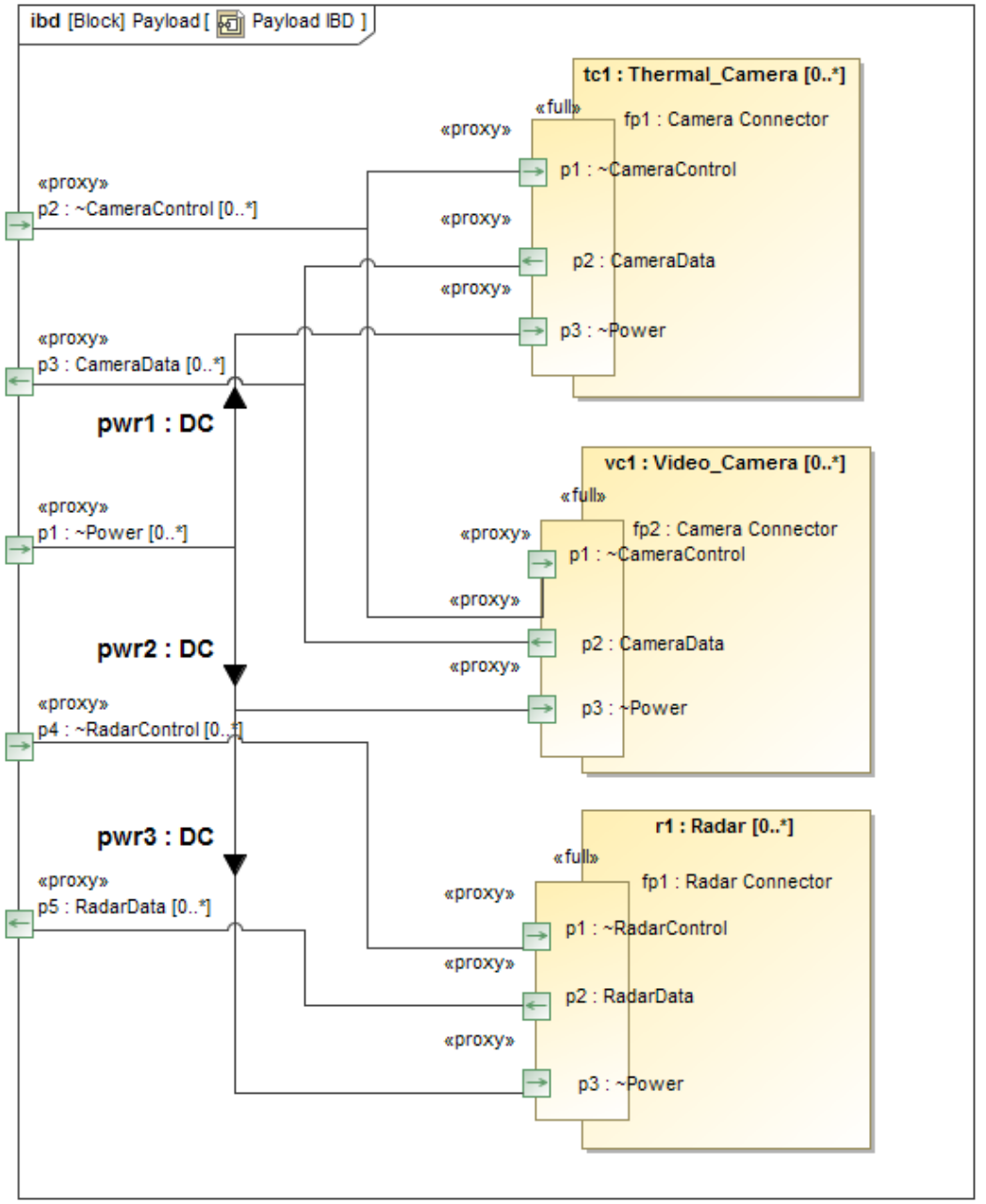
# Full Ports



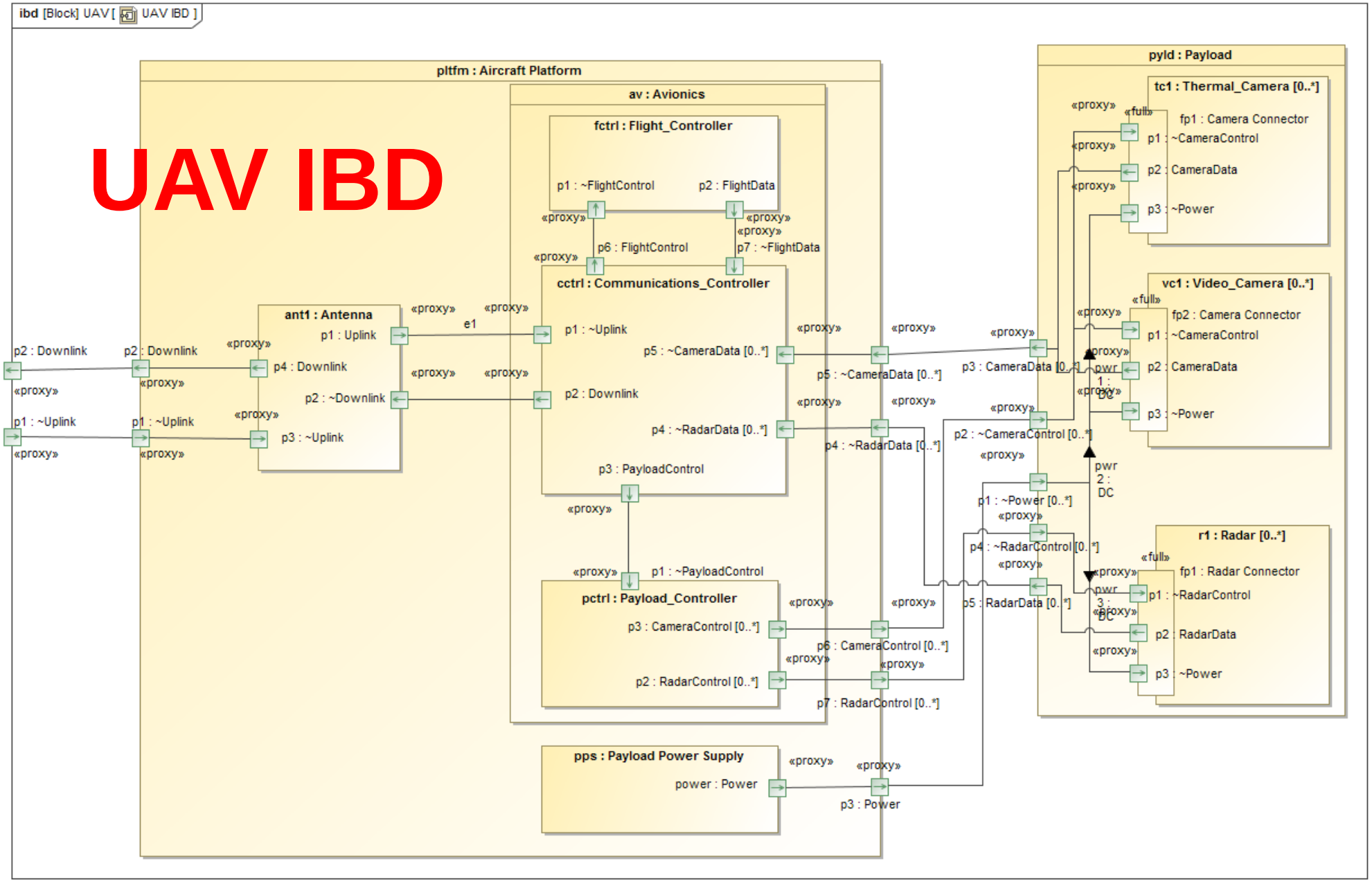
# Item Properties

- Item Flows can be Blocks or other types of Items
- But higher fidelity modeling uses Item Properties
- Item Properties are properties of the block that owns the IBD
- And are typed by an Item (a Block, Value Type, or Signal)
- They are associated with a Connector as an Item Flow
- They can be considered “Parts that Flow”, like water in a piping system or electricity in a circuit
- They allow the same Item to have different values in different parts of the system
- They can be very useful in analysis.

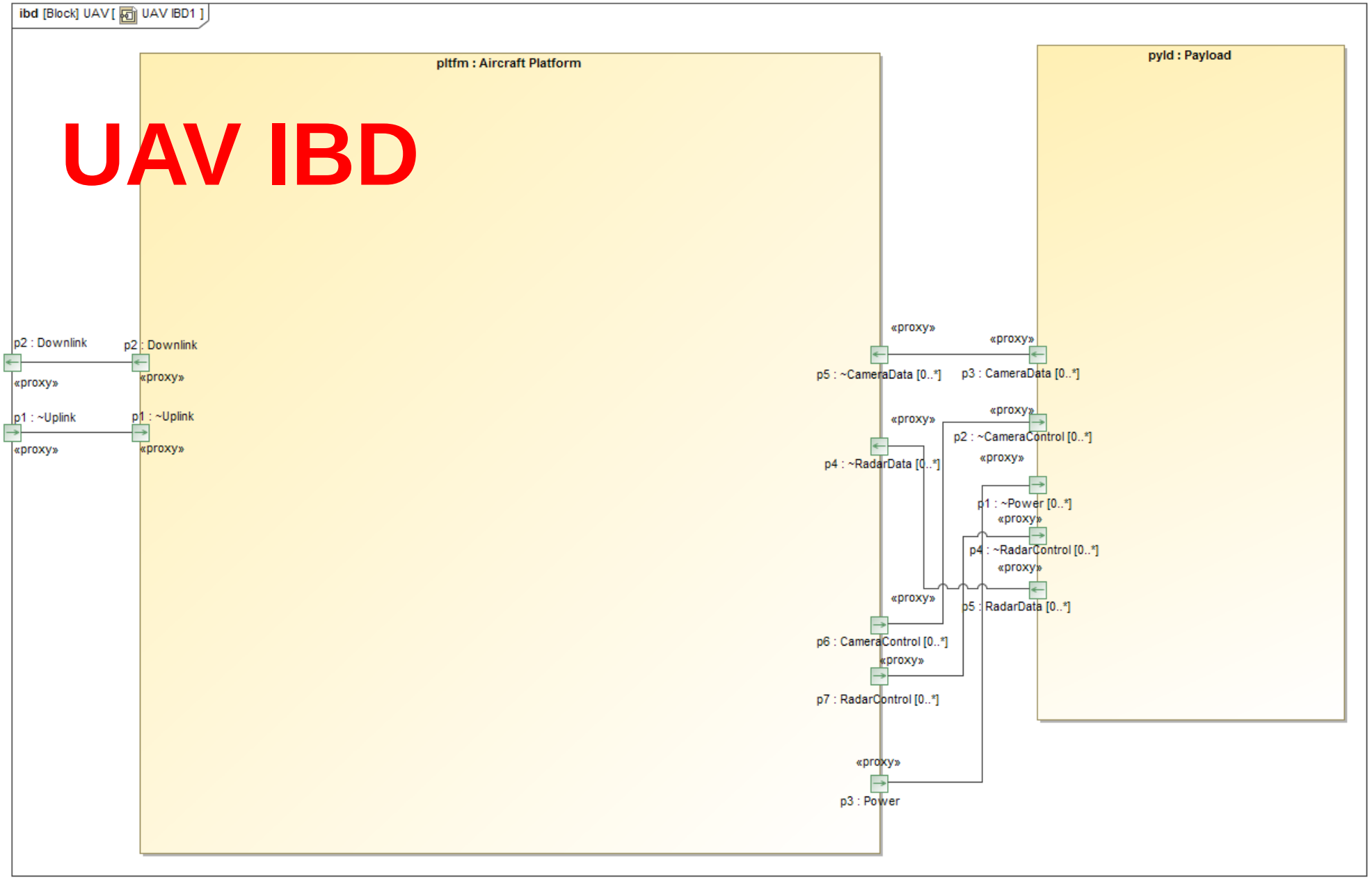
# Third Exercise



# UAV IBD



# UAV IBD



# Recap

- At the end of the hands-on exercises, you should be able to
  - Explain the following terms: connector, item,  item flow, proxy port, interface block, flow property, item property, full port
  - Create a set of connectors and item flows and display them in an internal block diagram
  - Add proxy and full ports to a block and use them in an IBD
  - Identify the principle purpose(s) of internal block diagrams

# Questions?

**www.intercax.com**

**email info@intercax.com**

**blog www.intercax.com/blog**

**Twitter @intercax | LinkedIn intercax-llc**

## Disclosure

- This course material is developed by InterCAX LLC. All copyrights held by InterCAX LLC.
- Usage of this material is governed by the licensing agreement terms between user and InterCAX. Usage or distribution for educational or other purposes outside these terms is not permitted. No part of this material may be reproduced or repurposed without the written consent of InterCAX.
- The intent is to present vendor-independent concepts and examples in an objective educational way that the course participants will find helpful. References are made to commercial products by InterCAX for the purpose of making these concepts concrete. Course participants are responsible to evaluate these products and tools for themselves and to investigate similar products and tools by other organizations where applicable.