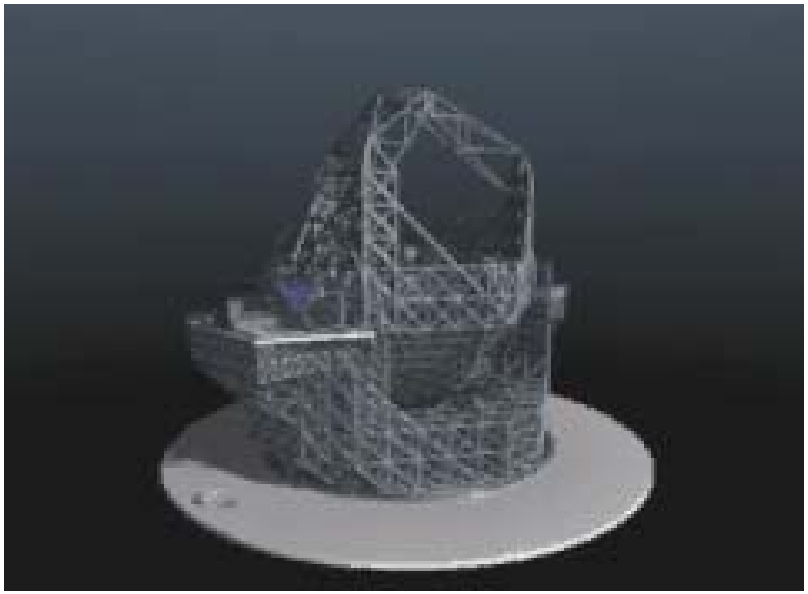
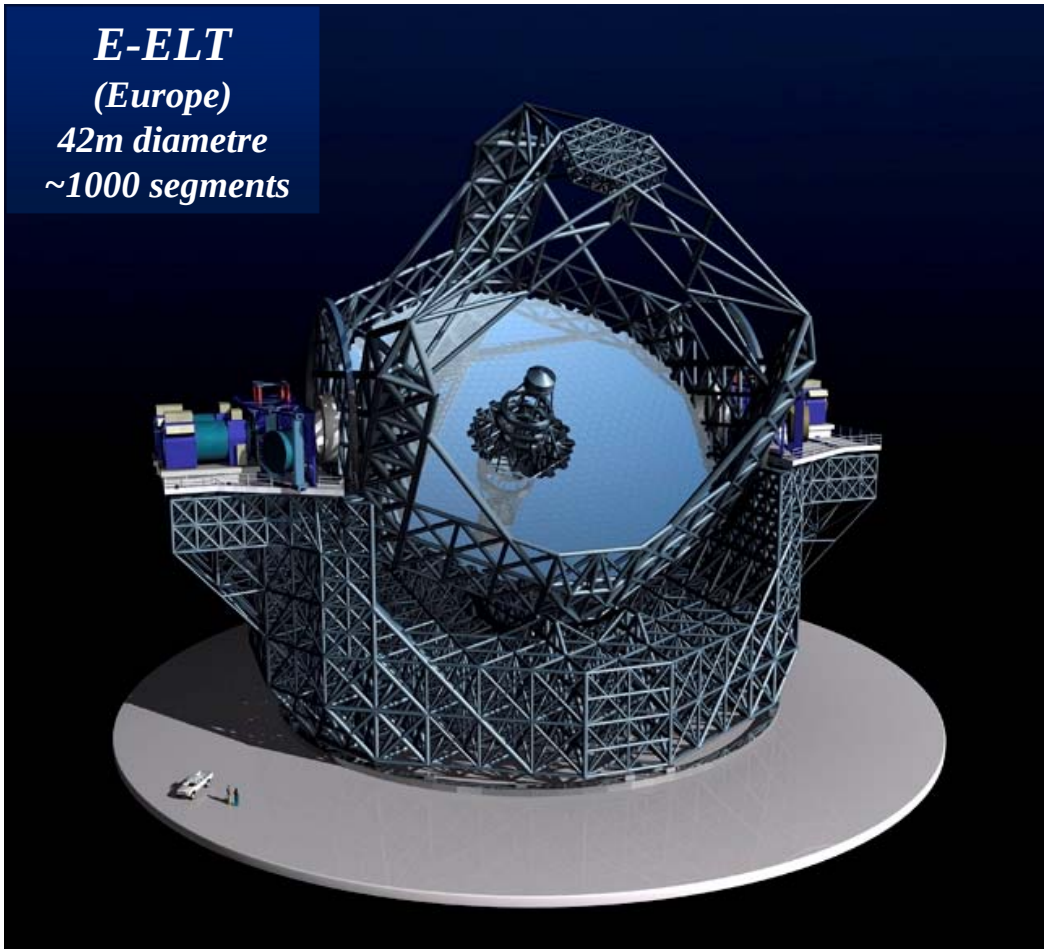
SysML for Telescope System Modeling

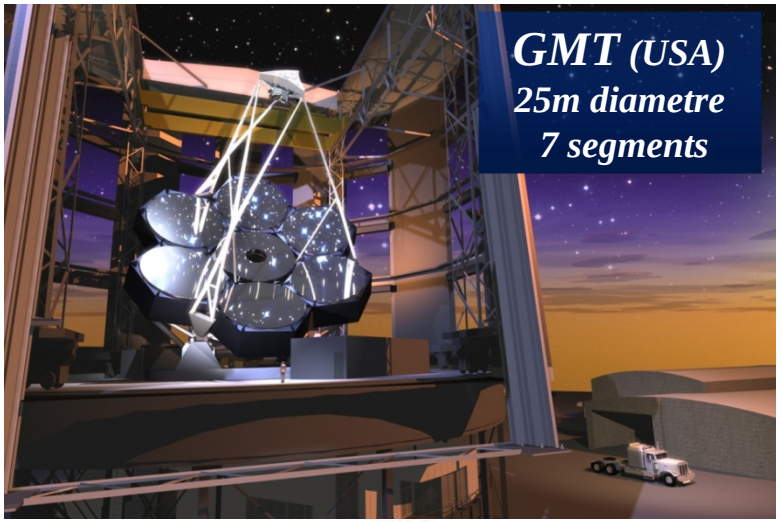


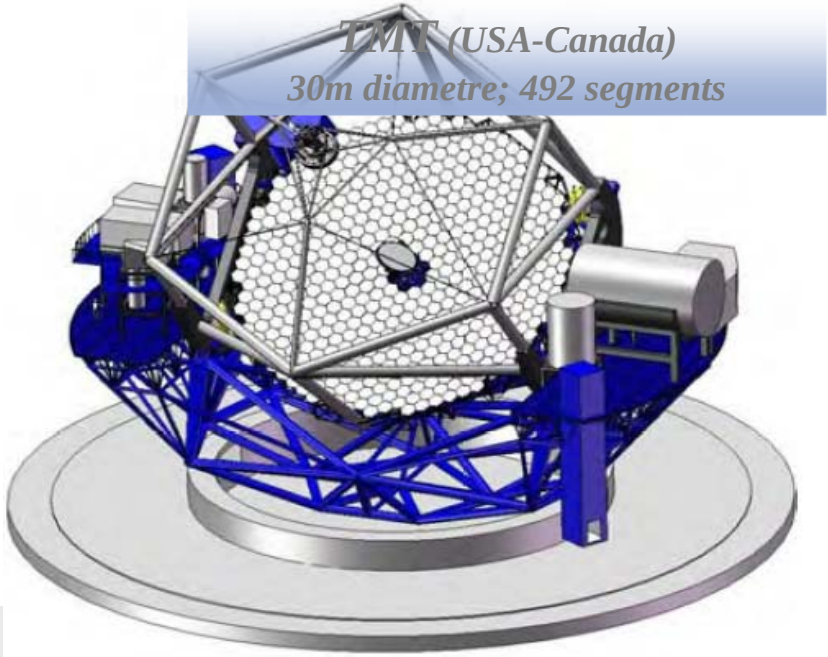
SysML for Telescope System Modeling Proceeding I – 2008-10-01

INCOSE MBSE Challenge Team SE^2
Robert Karban (ESO)
Tim Weilkiens (oose GmbH)
Andreas Peukert (TUM)
Rudolf Hauber (HOOD Group)

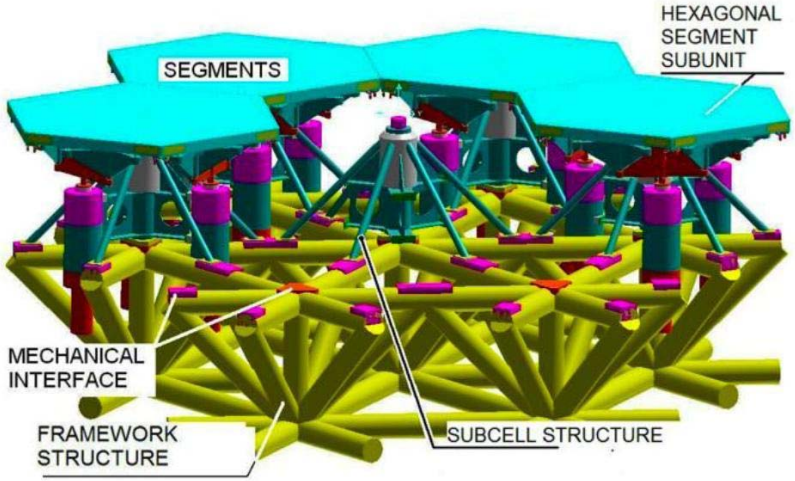
Images on this slide were produced by ESO

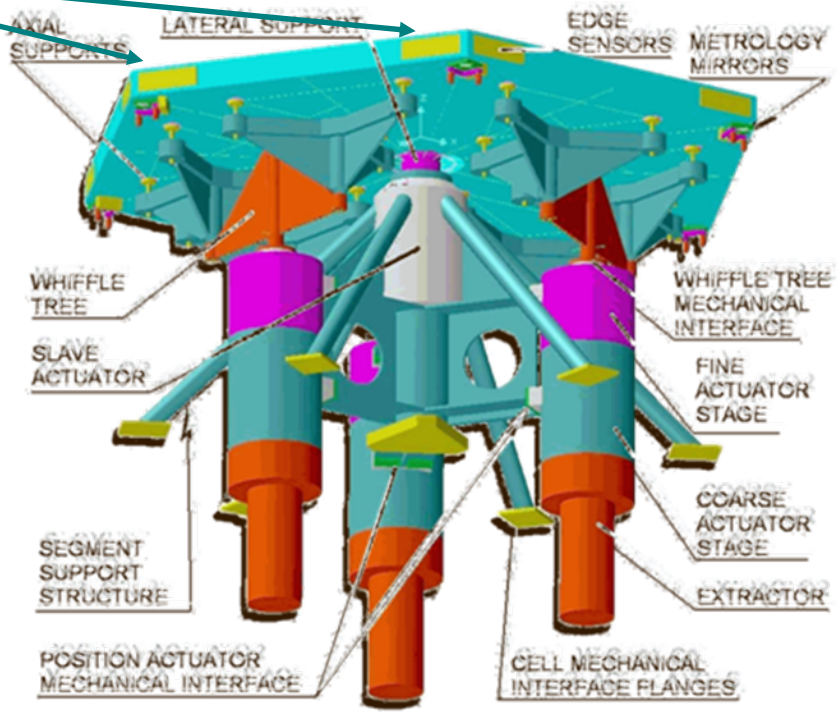




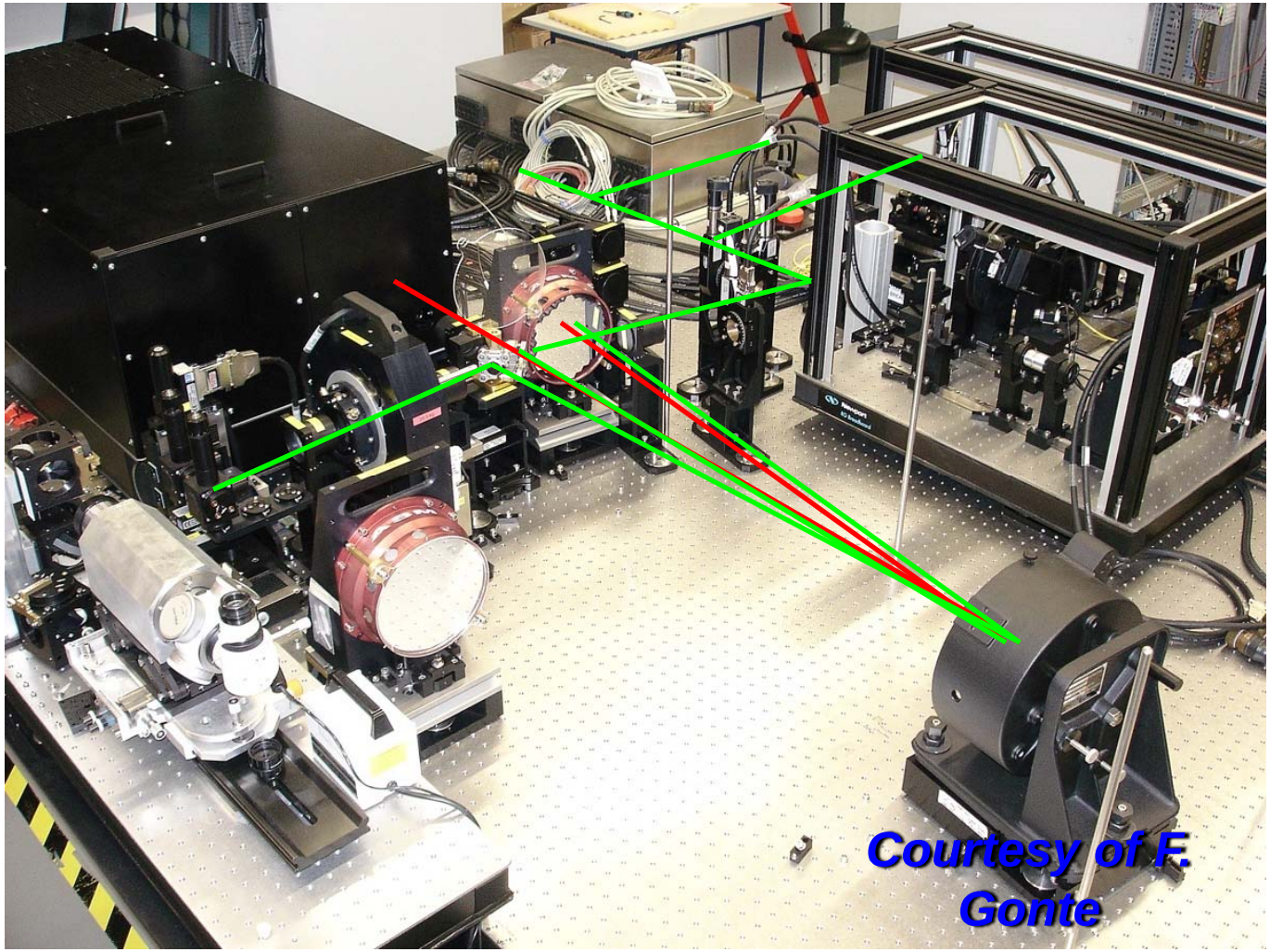


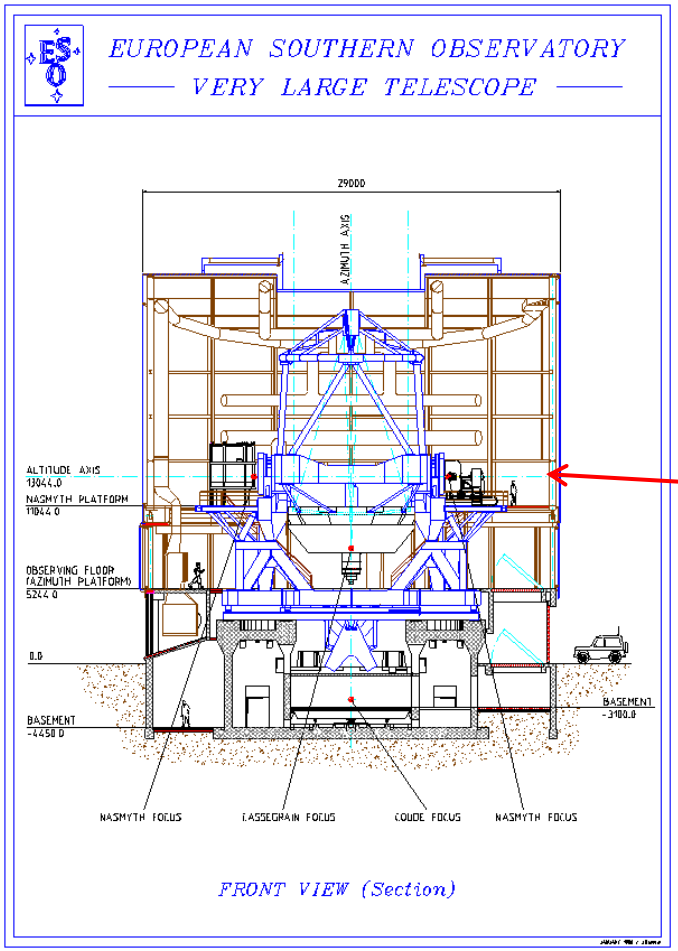
Edge Sensors

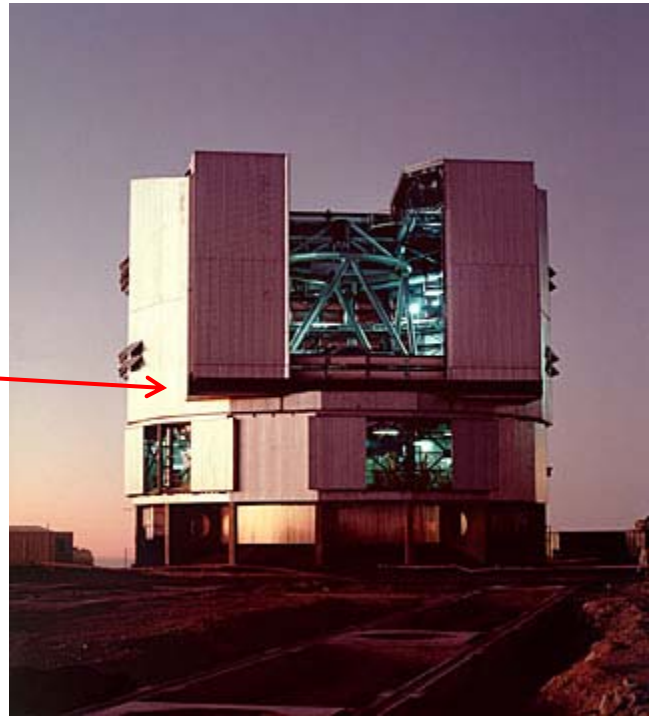




Detect nanometers of phasing error in micrometers of turbulence with Phasing Wave Front Sensors (~20 nm RMS)







APE will be installed at the telescope in the Chile desert.

Deliverables:
Generic SysML modelling FAQ: Excerpt 1/2

- General modeling guidelines
  - How should I name model elements?
  - What rules should I follow when creating diagrams
  - How should I document the model?
  - How do I use different types of annotations in the model?
  - How should I structure the model by using packages?
  - How do I include external references?
- Guidelines for necessary system models and aspects
  - What system views should my (structural) model contain?
  - How many levels of abstraction do I need?
- Guidelines for modeling the system requirements
  - How should I use dependency matrices?
  - How do I model relationships between requirement and design element?

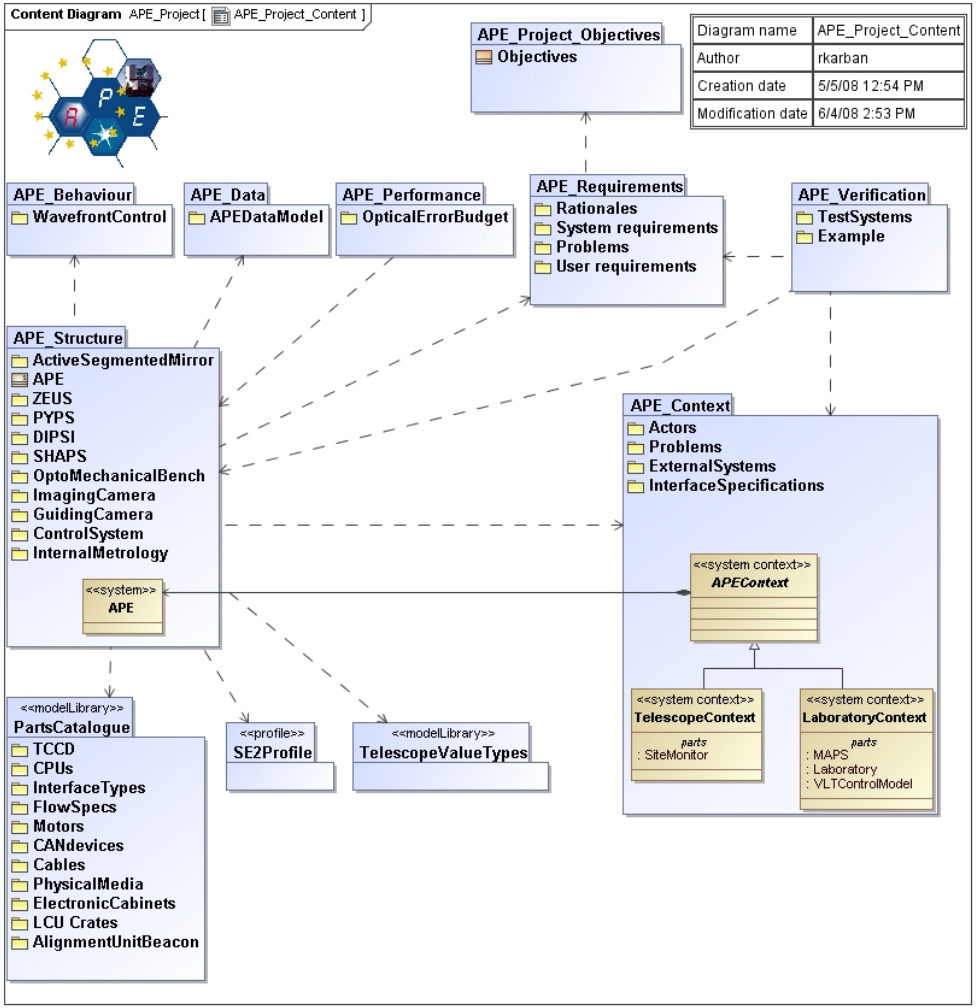
Deliverables:
Generic SysML modelling FAQ: Excerpt 2/2

- Guidelines for modeling the system structure
  - How do I distinguish a sub structure and an assembly?
  - How do I model different contexts?
  - Where do I put systems which are part of the project and needed in different contexts but nor part of the system itself?
  - When should I use block, data or value types?
  - How do I model re-usable parts, like a catalogue of building blocks?
  - Where do I put (new) domain specific model elements, like stereotypes?
  - How do I model domain specific values and types?
  - How do I model design variants?
  - How do I define system hierarchies?

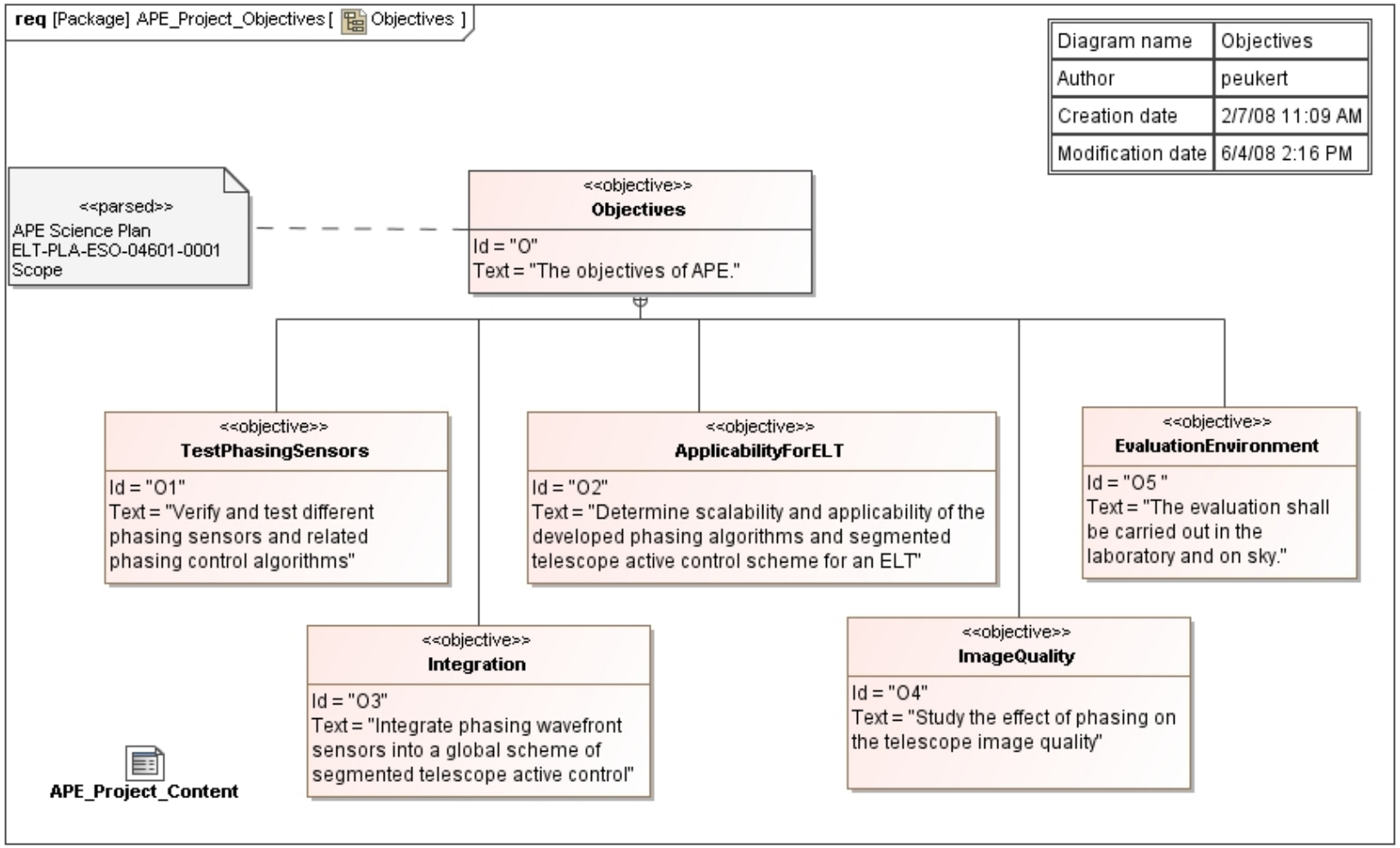
Deliverables:
SysML model for the APE project

- Three major model parts:
  - Actual system model: APE (with all mentioned system aspects)
  - Catalogue model: standard parts, library of block prototypes
  - Modelling profile: additional stereotypes
- Main characteristics:
  - Scalable model structure and organisation
  - Includes model annotations, external references
  - Various examples of ports and flows to model interfaces
- Abstraction levels
  - Functional, Structural, Deployment
- Preliminary results are available at **http://mbse.gfse.de**

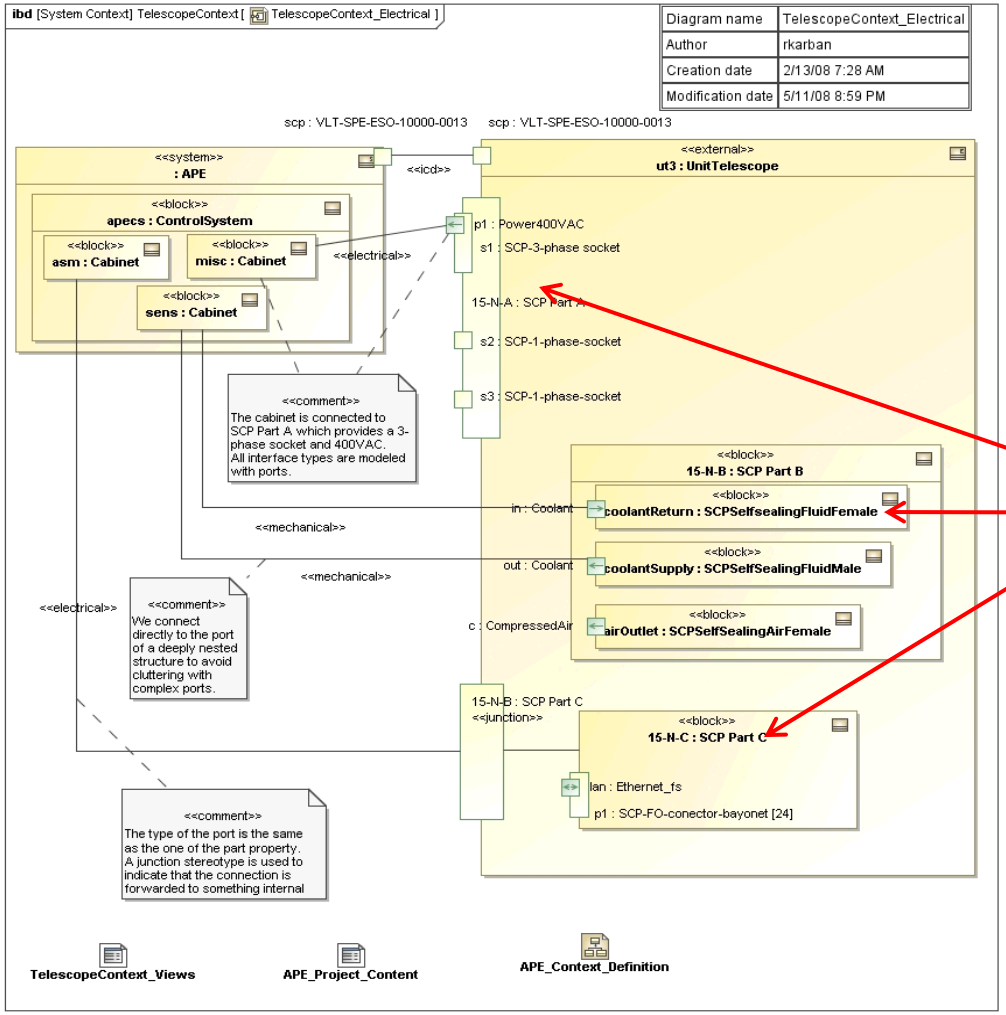
APE project : Overview



APE project: Objectives / requirements



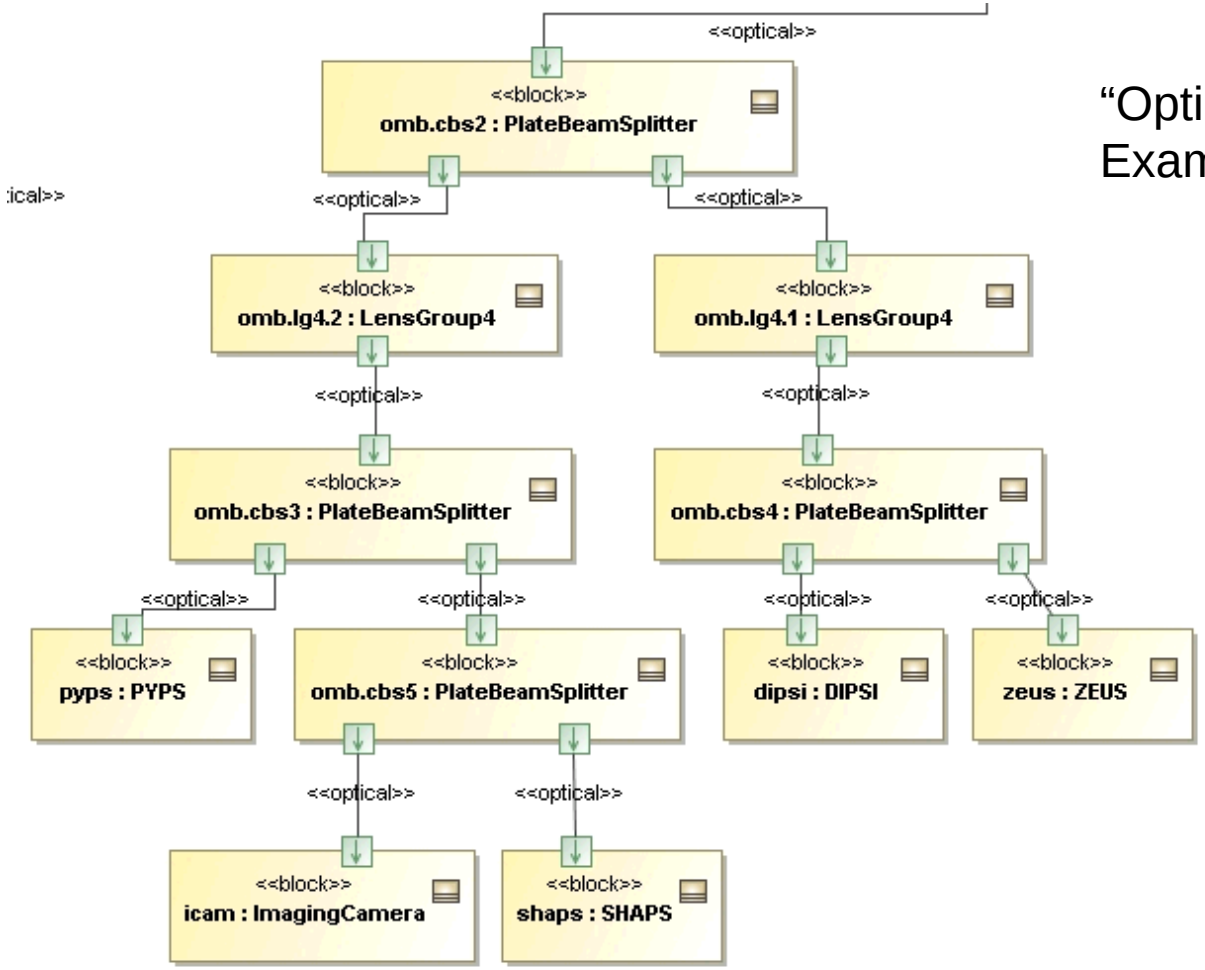
APE system model: System context



3 modeling approaches for interfaces

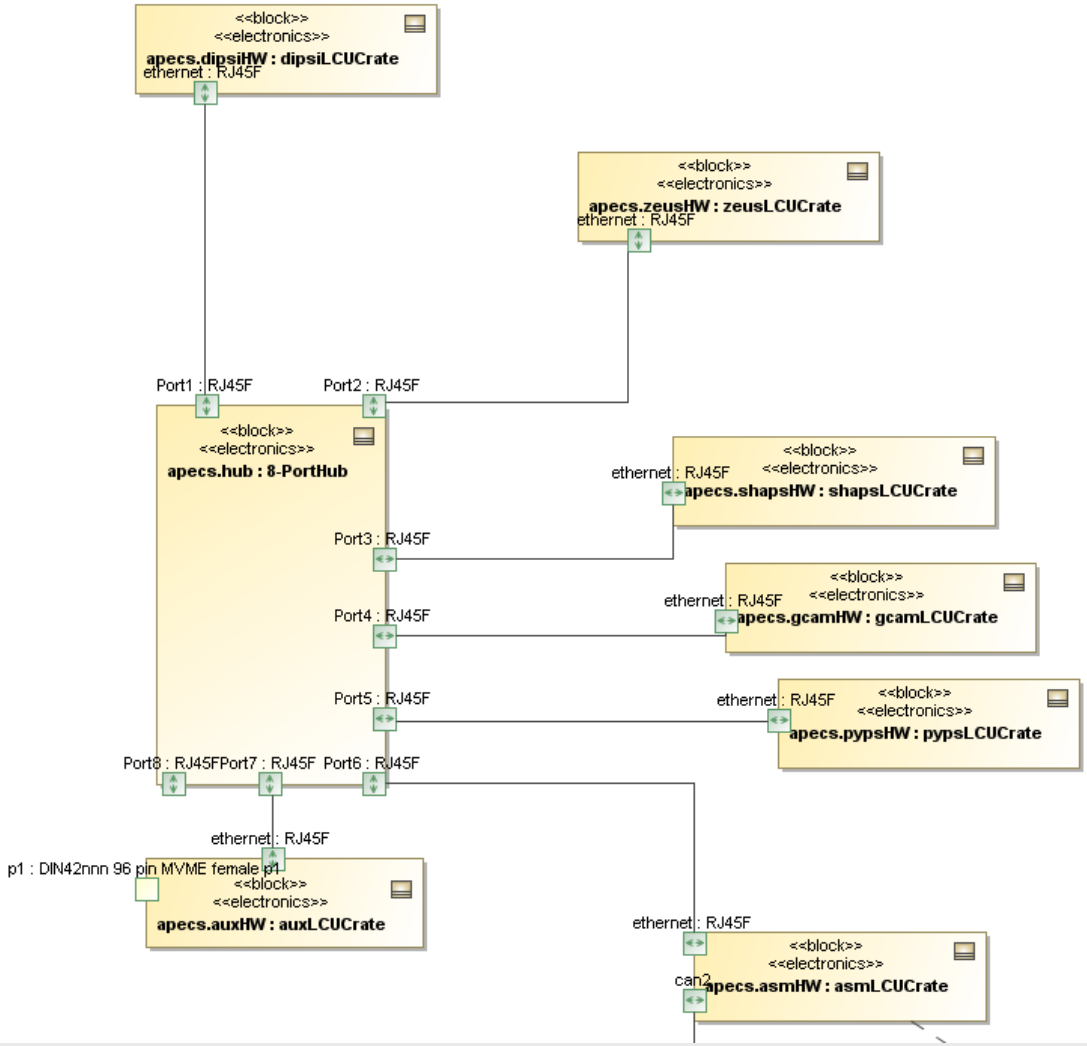
➔ treated later in challenges

APE system model: Structure: Internal structure



"Optical view" of APE:
Example for using nested parts

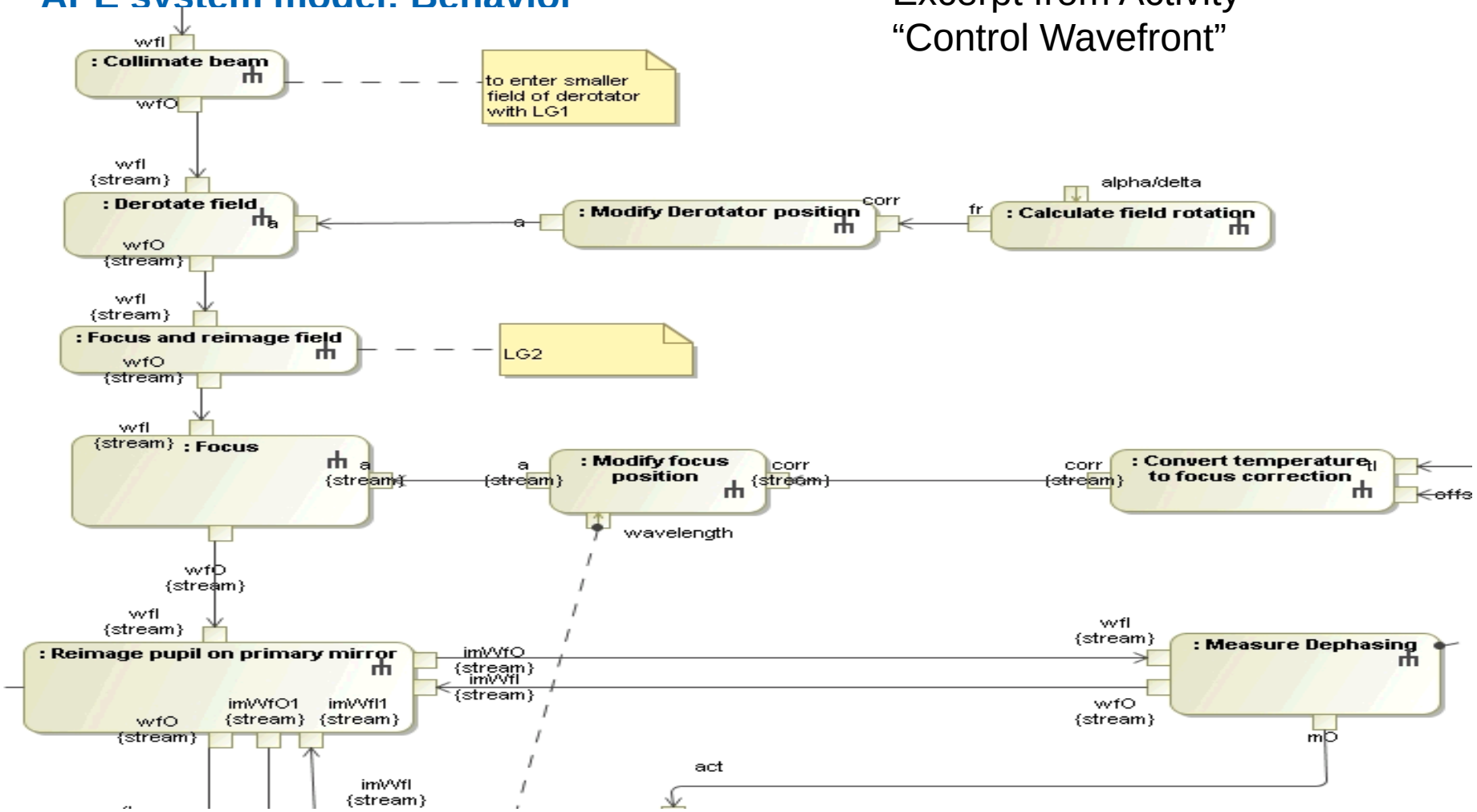
APE system model: Structure: Internal structure



"Electrical view" of APE

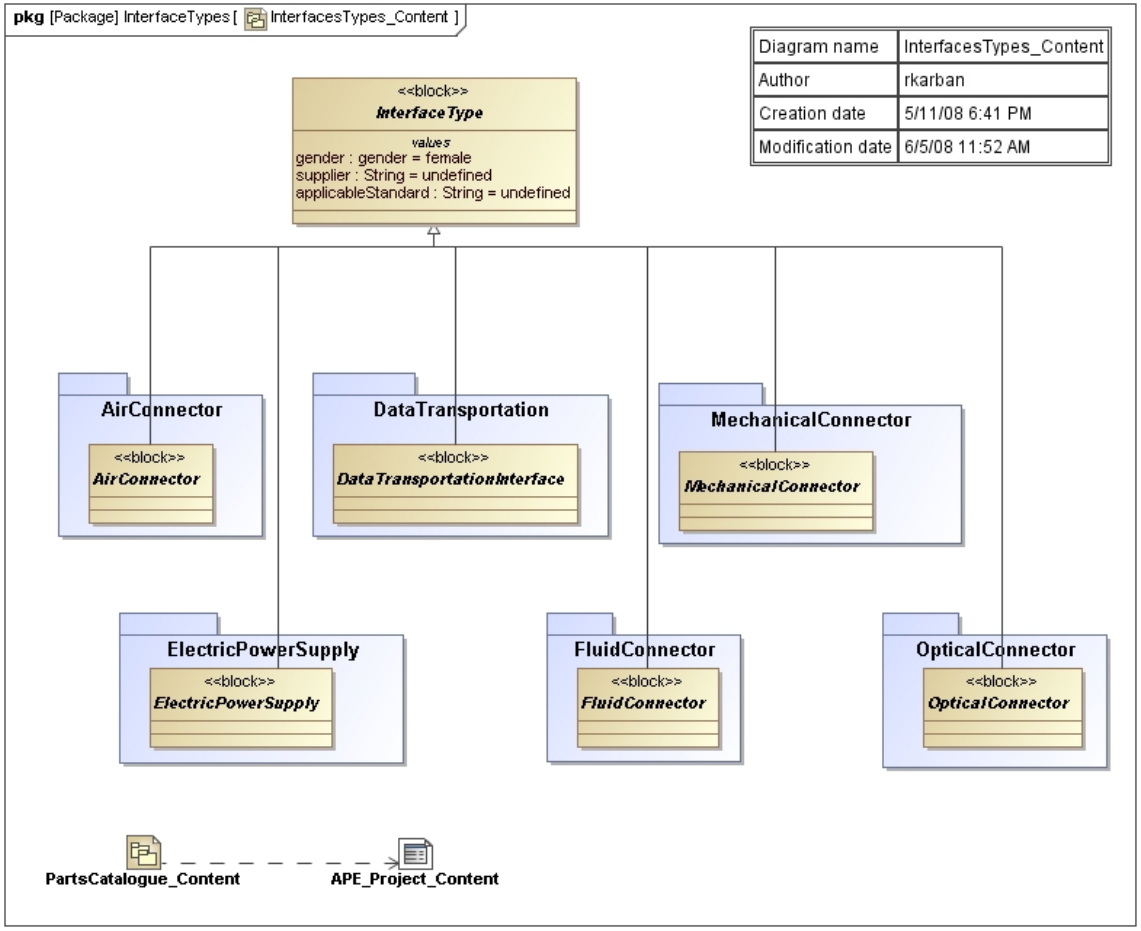
APE system model: Behavior

Excerpt from Activity
“Control Wavefront”



Catalogue model: Abstract types

Example for catalogue:

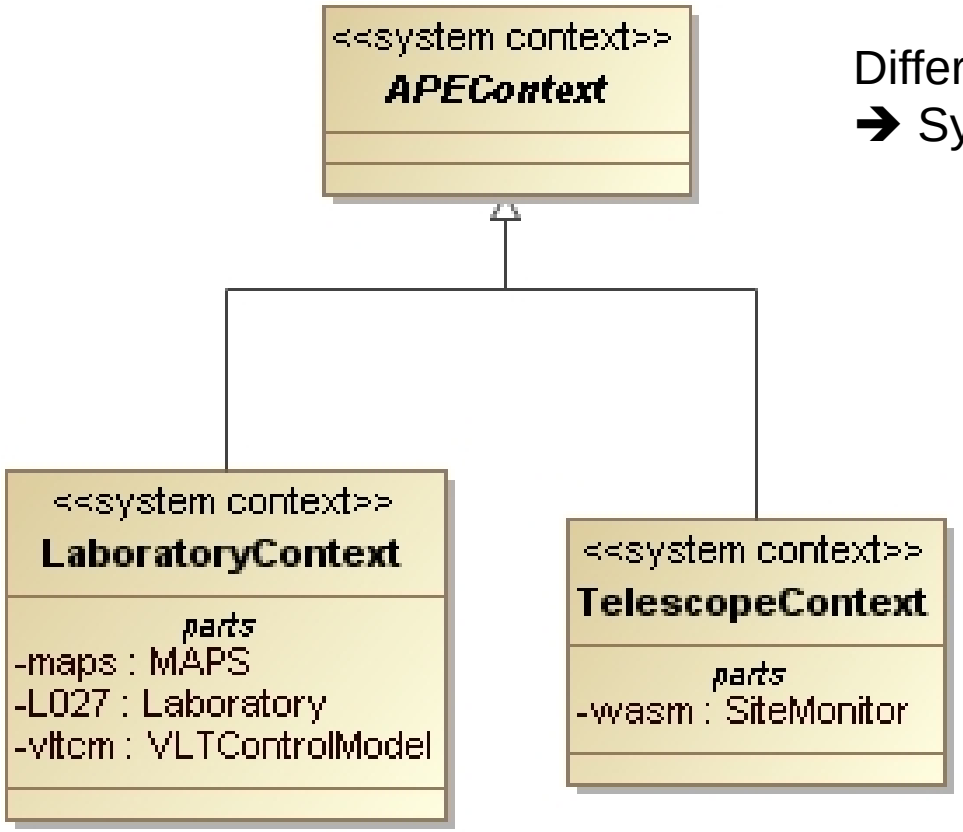


Challenges identified at IS08

- **Variant modeling**
- **Connection of nested blocks**
- **Grouping of interfaces with nested ports**
- **Logical vs. Physical decomposition**
- **Functional multi-layer abstraction**
- **Reuse of blocks, allocation and instances**
- Structural multi-layer abstraction
- Defining Quality of Service (QoS)
- Transition to UML for software
- Usage of name spaces
- Configuration Control
- Navigability

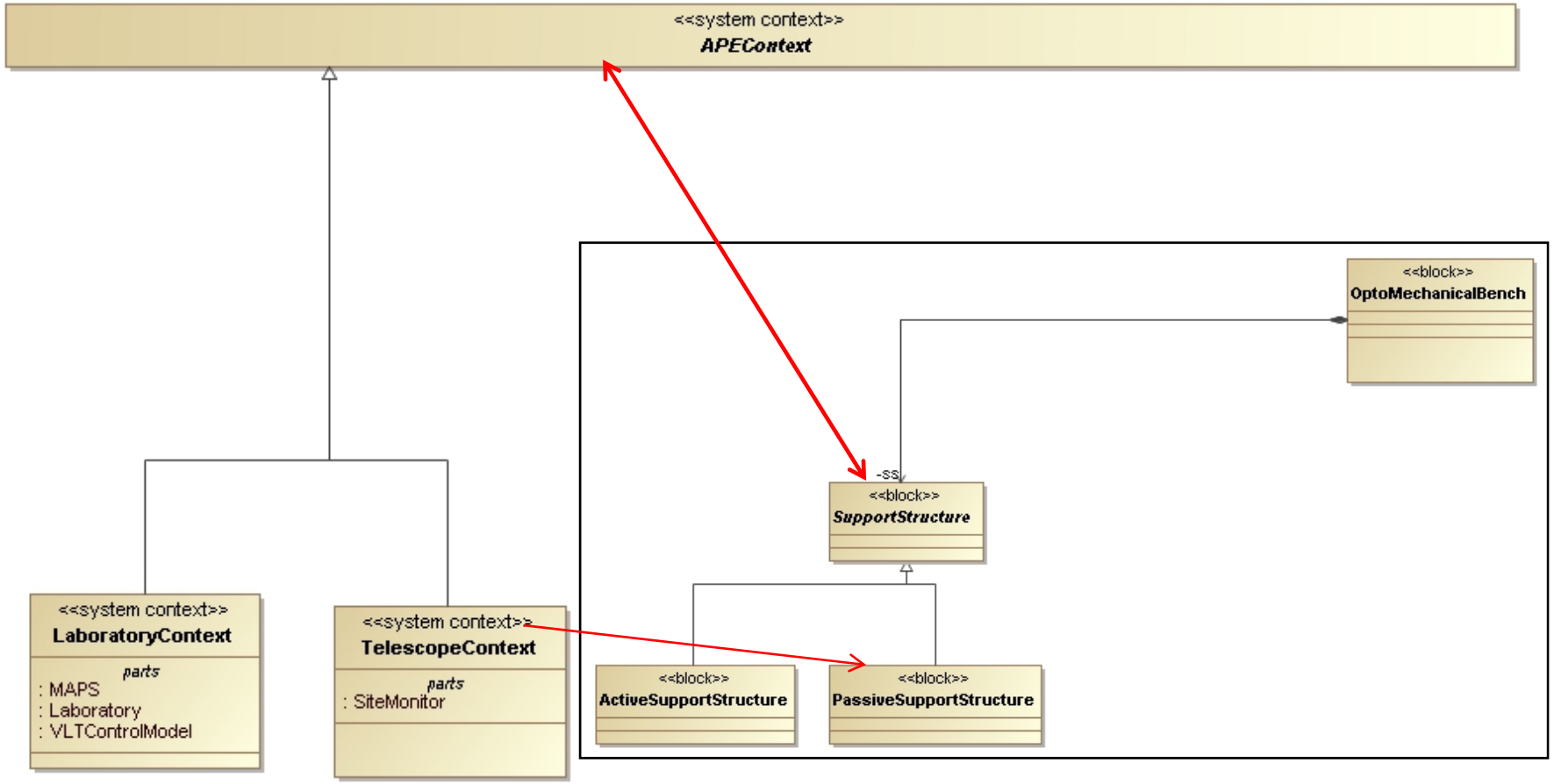
Note: *Order has no meaning, e.g. priority*

SysML challenge: Variant modeling – Example (1/2)

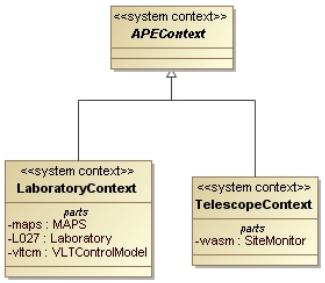


Different contexts imply different design
➔ System variants

SysML challenge: Variant modeling – Example (2/2)



SysML challenge: Variant modeling – Characteristics



Notion

- It‘s an objective of SysML to support evaluation of different system
- Variants are common in system modeling, e.g. different deployment, allocation, connection, parts

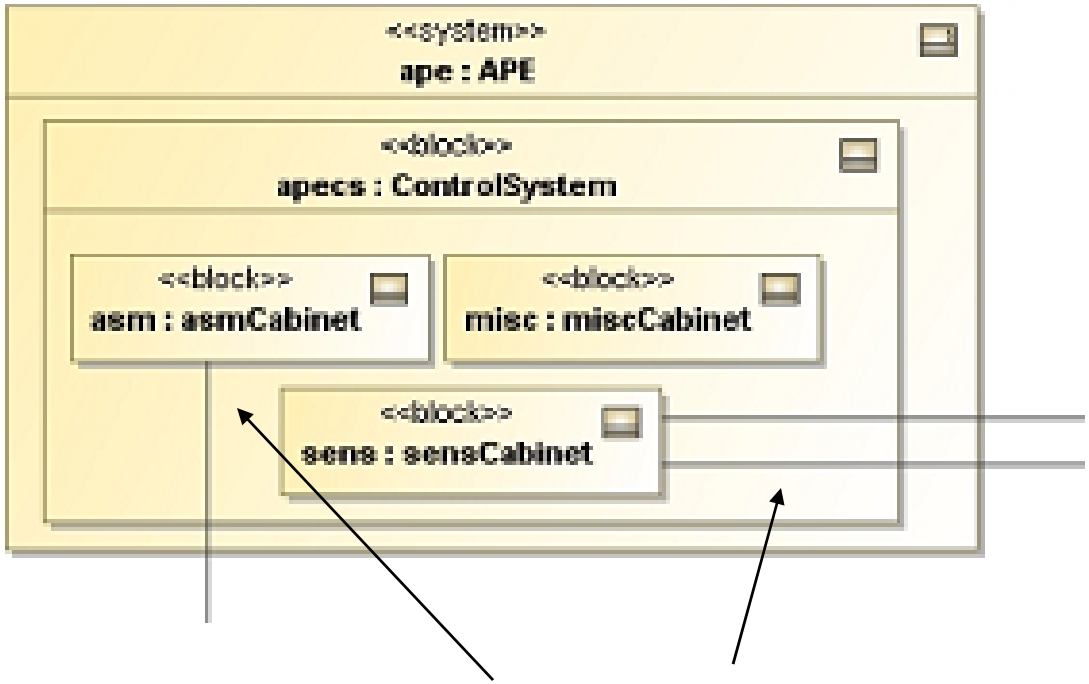
How to

- Generalization
- Profile with stereotypes for variants.
- Tags for identifying parts and associated variants

SysML status

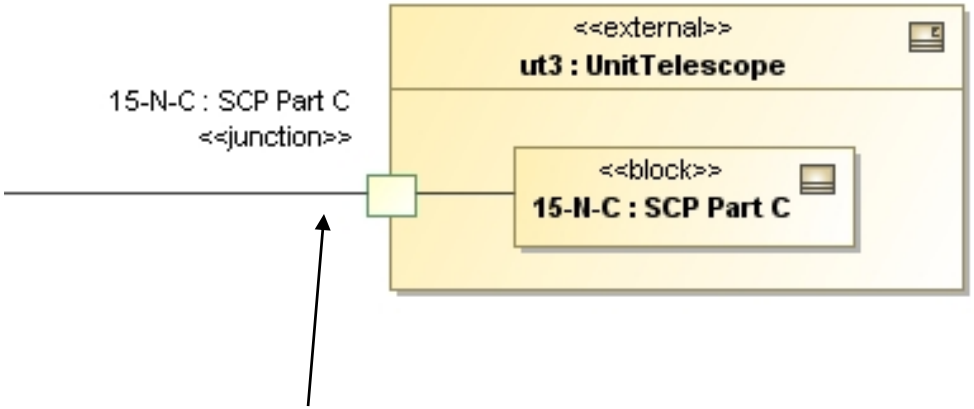
- Variant modeling is not officially supported by OMG SysML 1.1.
- Intentionally left out in OMG SysML.
- Planned feature for OMG SysML 2.0.

SysML challenge: Connection of nested blocks – Example (1/2)

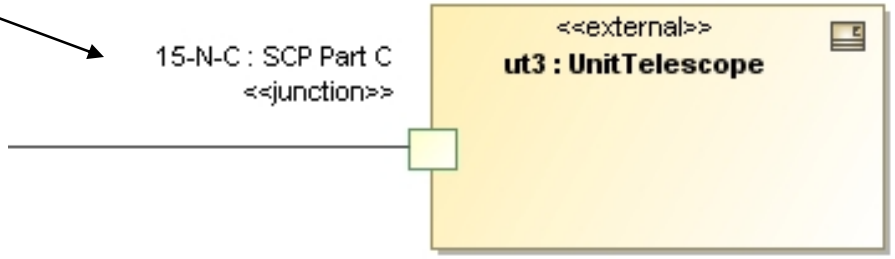


Nested blocks are connected directly with blocks outside.
Problem: Hiding internal blocks hides also the relationships.

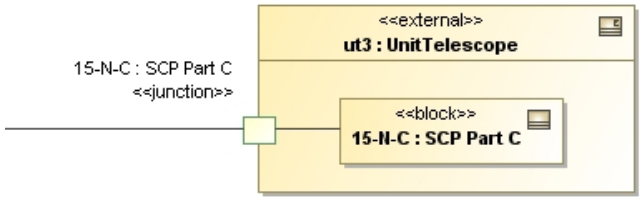
SysML challenge: Connection of nested blocks – Example (2/2)



Use a junction port as a proxy for the internal block.
Hiding nested blocks doesn‘t effect the relationships.



SysML challenge: Connection of nested blocks – Characteristics



Notion

- It must be possible to hide nested blocks without loosing the view on their relationships to external blocks.

How to

- Use ports with stereotype as a proxy for a nested block.

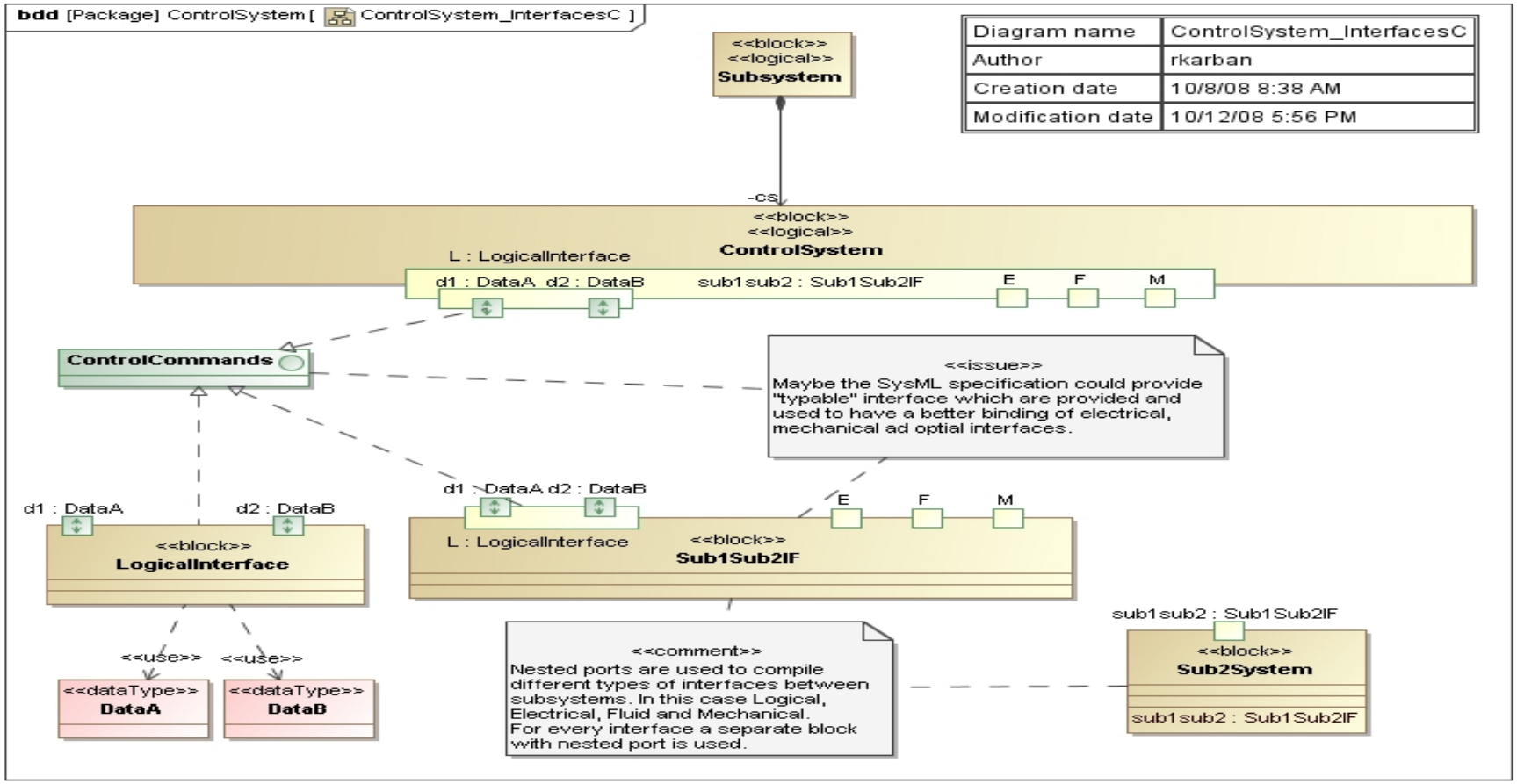
Problem

- Text of stereotypes clutters the diagram. Symbol for junction port is needed

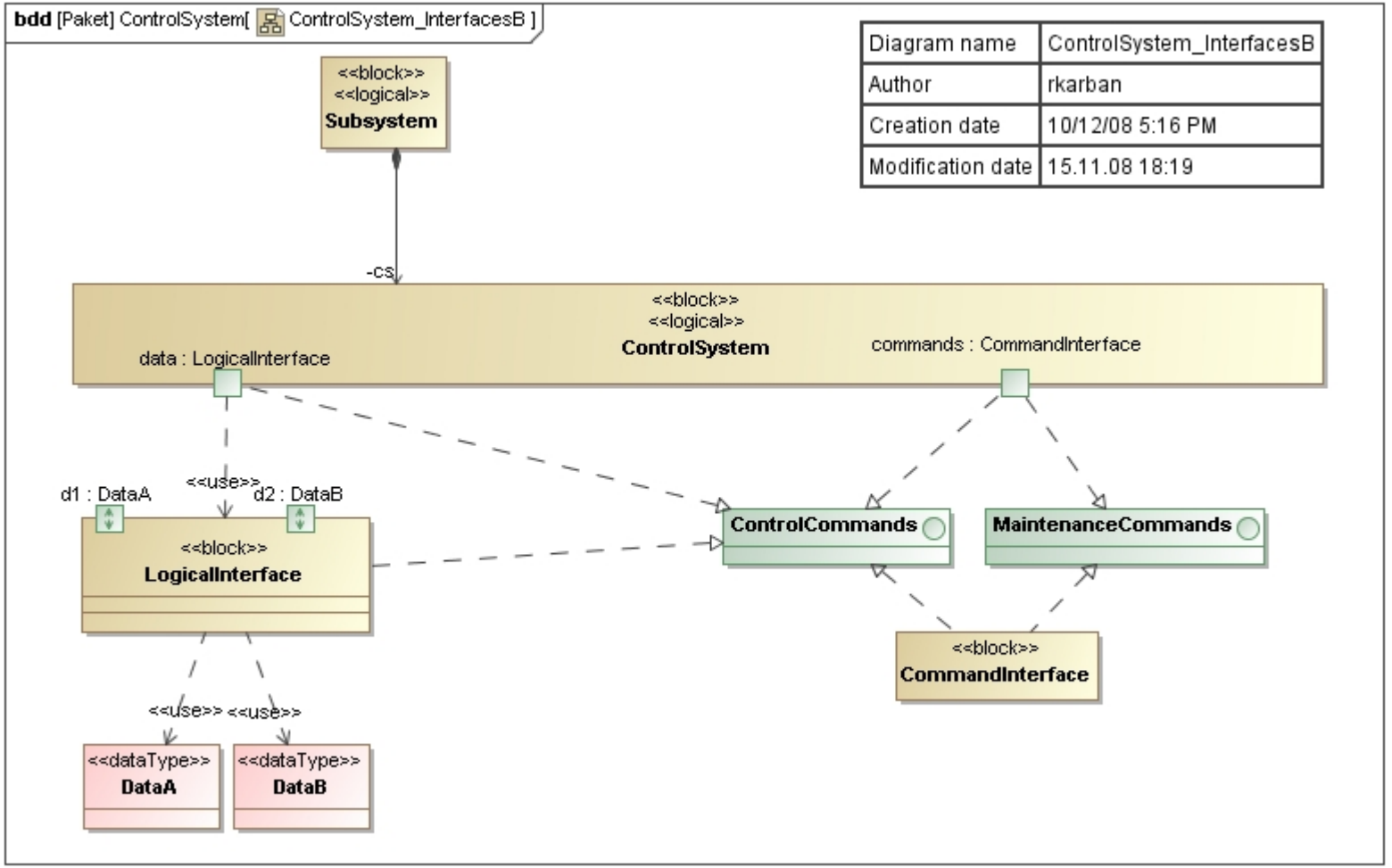
SysML status

- The probelm is recognized by the SysML working group.
- Issue for SysML 1.1, but deferred for future versions.

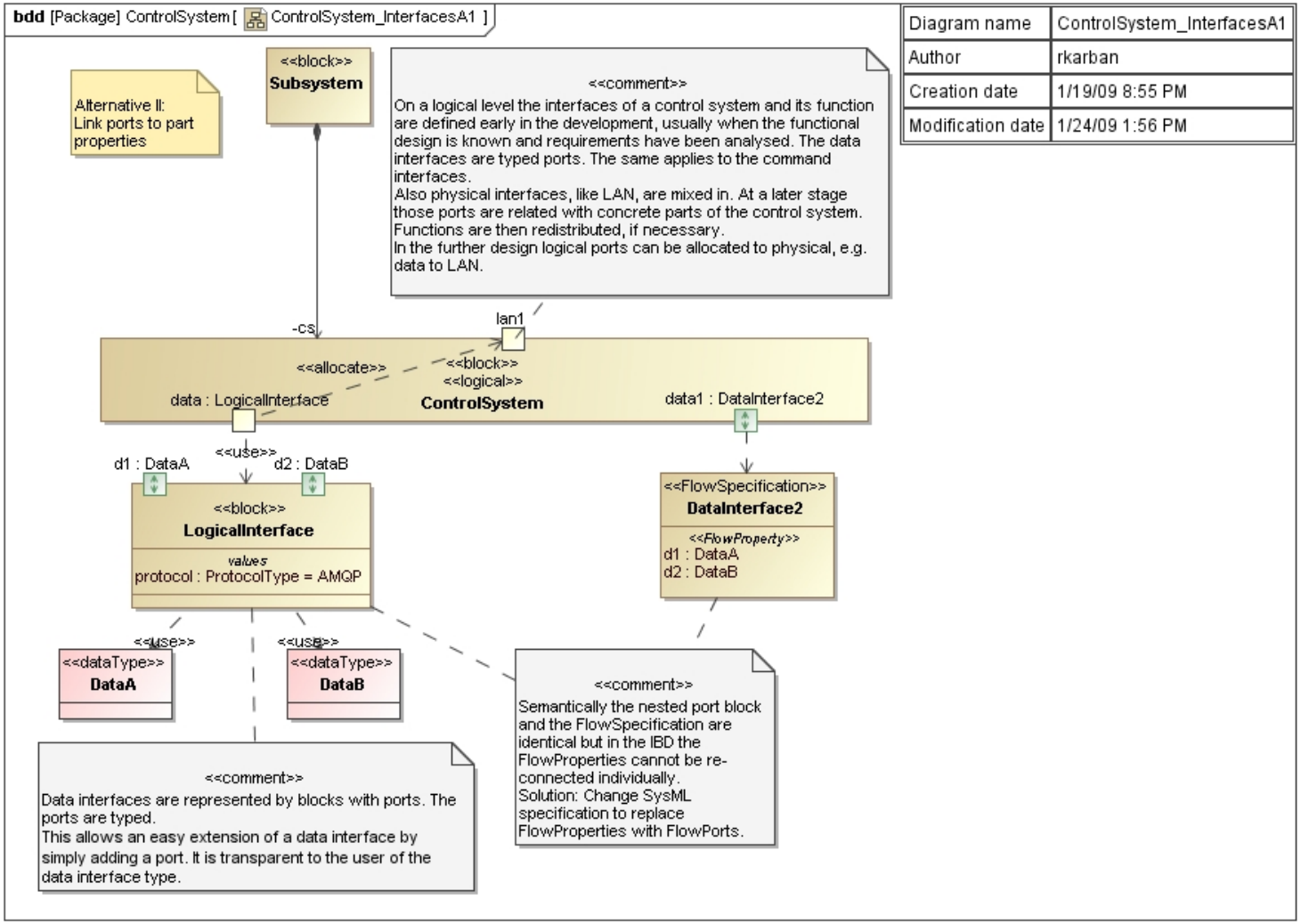
SysML challenge: Grouping of interfaces with nested ports (Example 1/5)



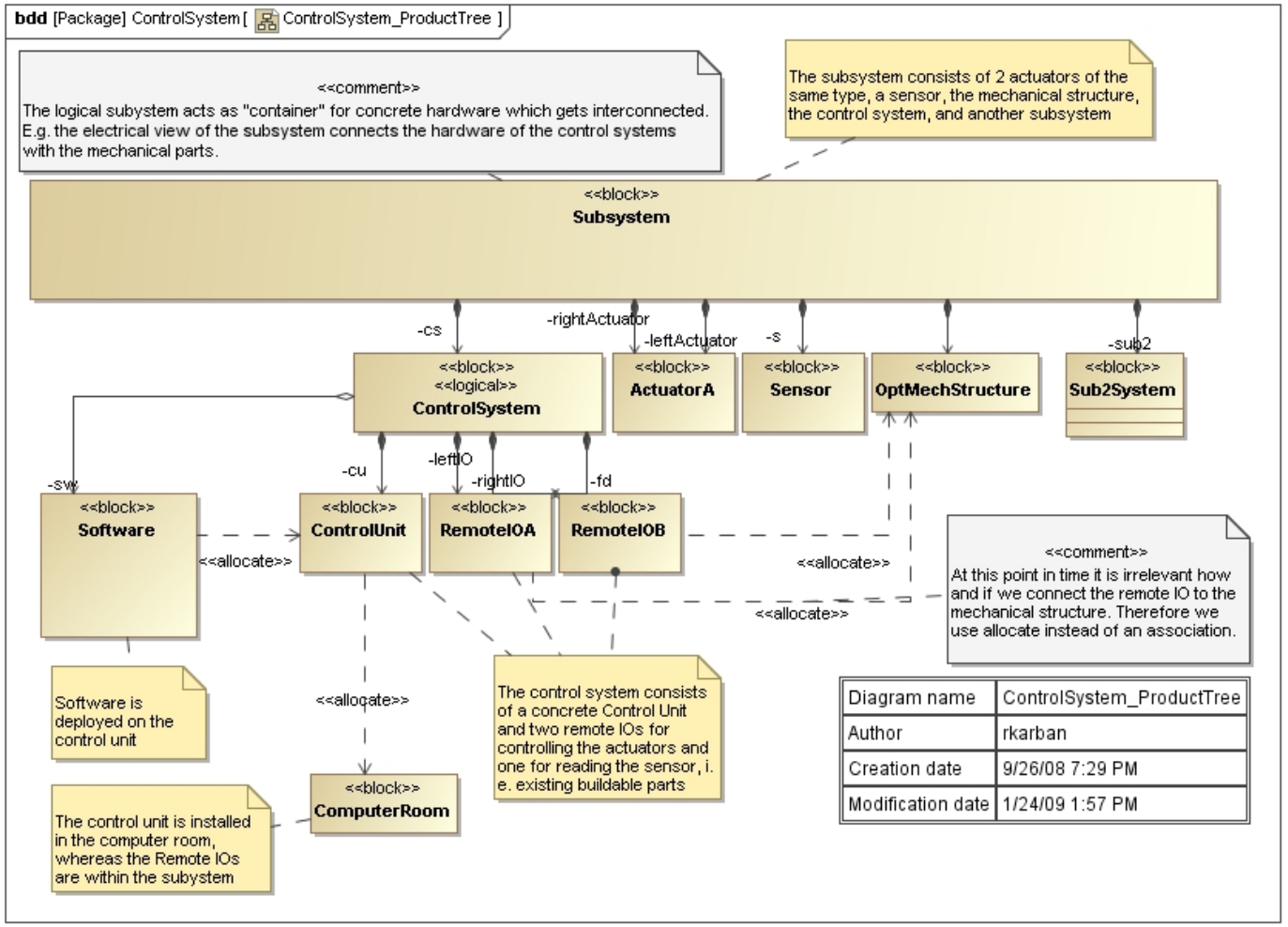
SysML challenge: Grouping of interfaces with nested ports (Example 2/5)



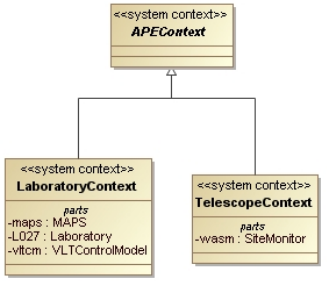
SysML challenge: Grouping of interfaces with nested ports (Example 3/5)



SysML challenge: Grouping of interfaces with nested ports (Example 4/5)



SysML challenge: Grouping of interfaces with nested ports – Characteristics



What to do

- Identify and describe unambiguously interfaces between systems, namely the provided and required part.
- Relay connections from outside to parts inside

How to do

- Create a nested block, which types a port, by grouping ALL interfaces between two components.
- Use FlowSpecifications to model nested, reusable flows

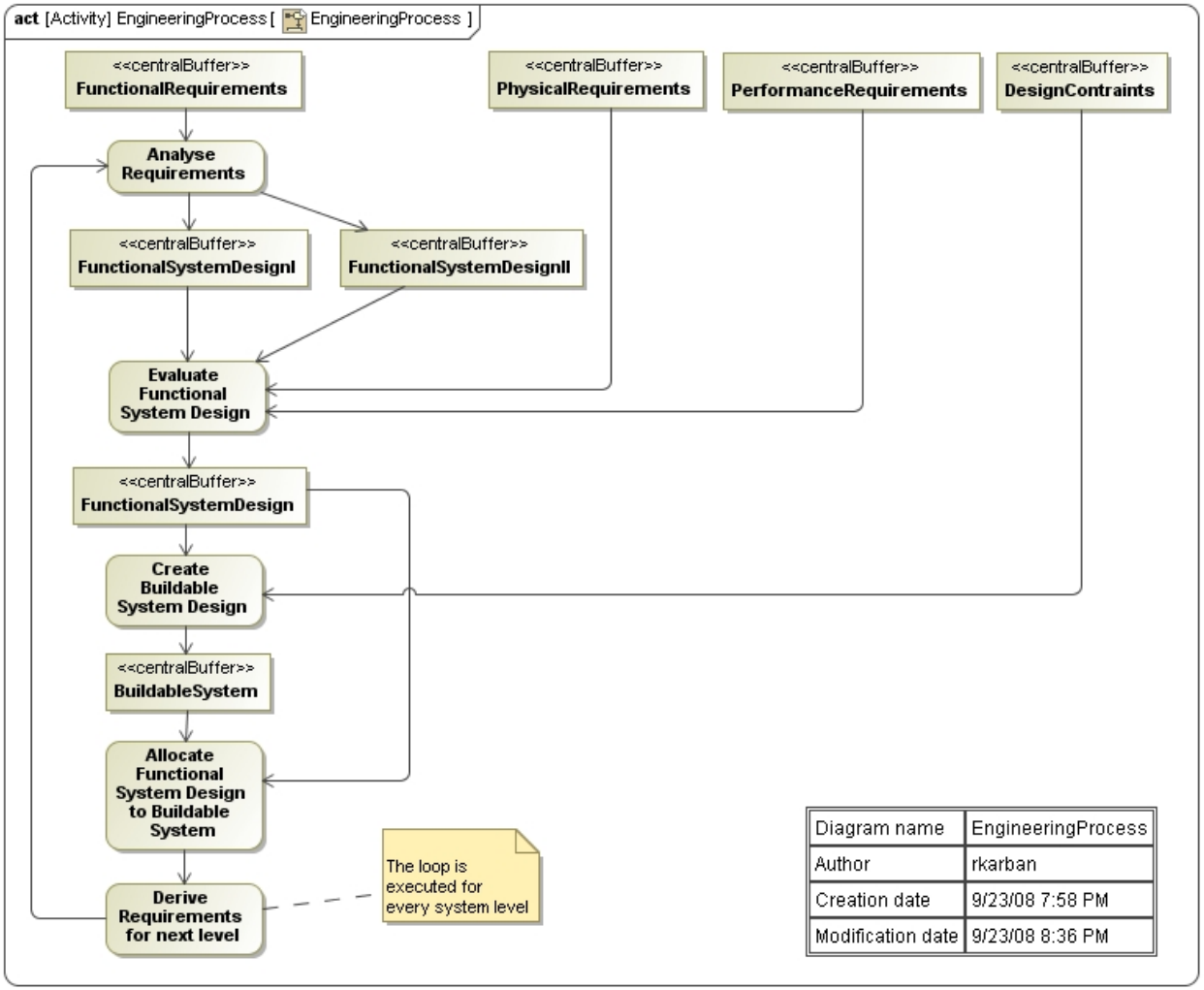
SysML issue

- FlowProperties cannot be connected in IBDs

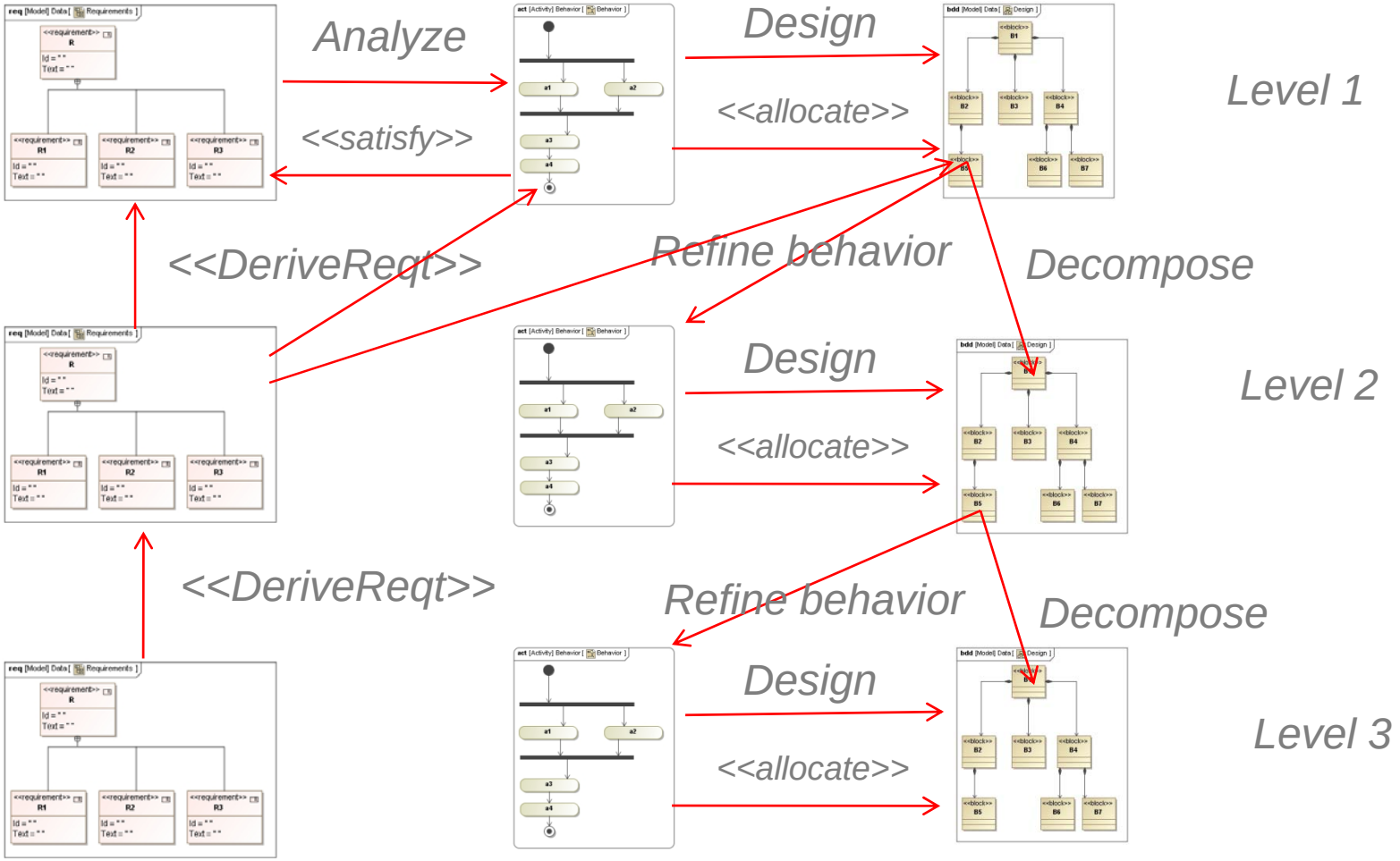
SysML status

- A request to the RTF will be submitted to enhance the definition of FlowSpecifications

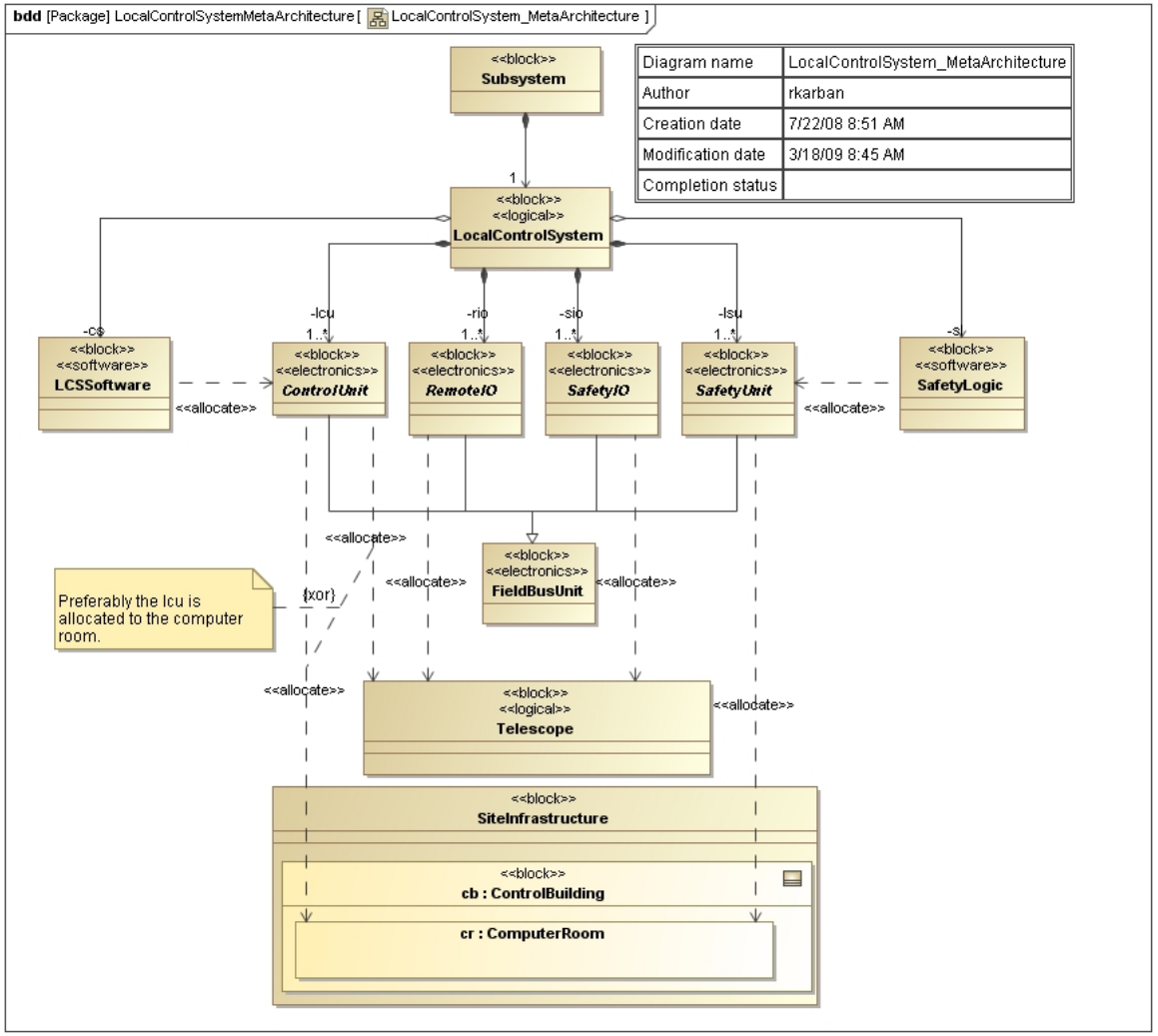
SysML challenge: Logical vs. Physical decomposition – Wymore like



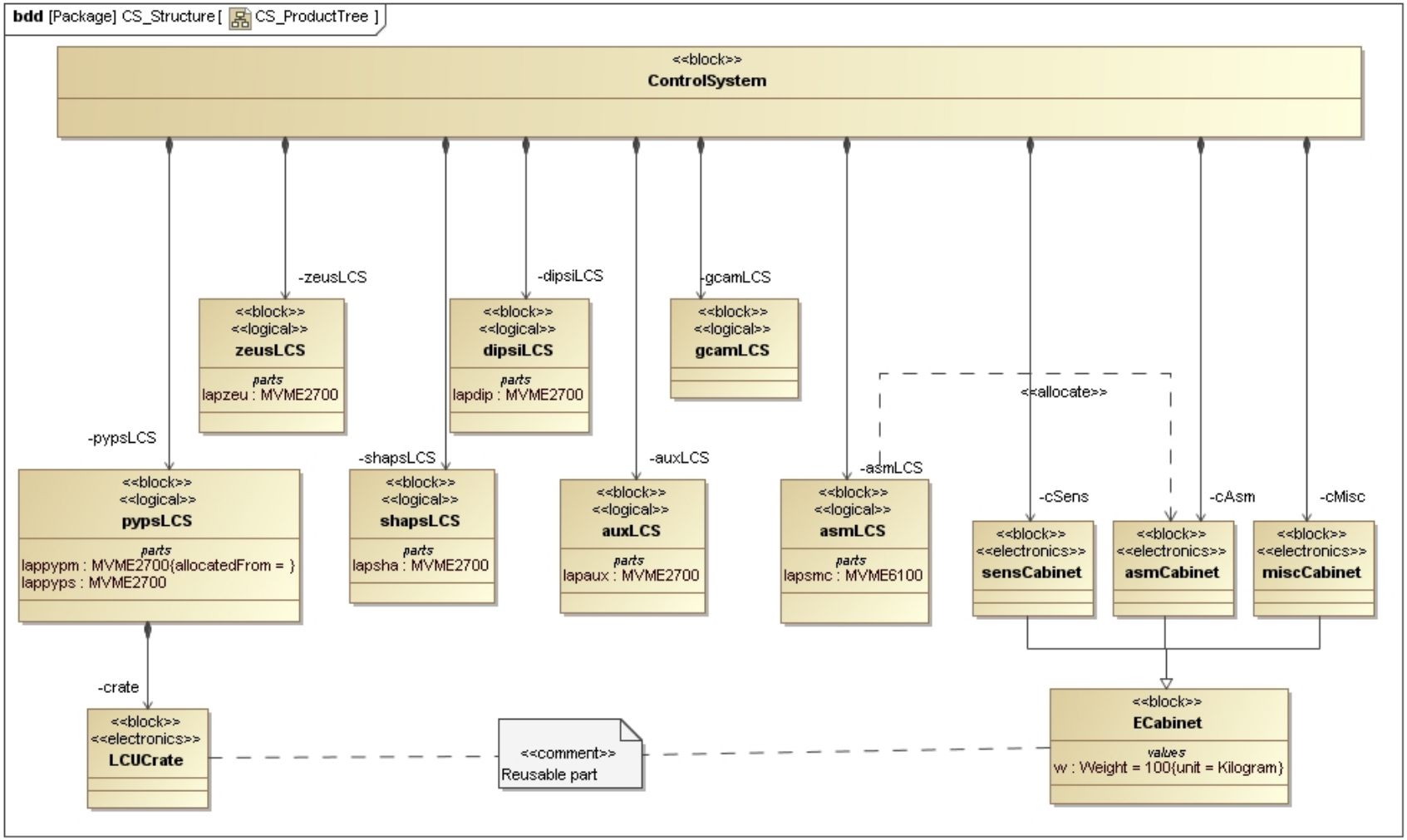
SysML challenge: Iterative Logical to Physical multi-layer allocation



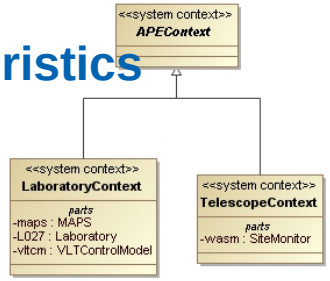
SysML challenge: Logical vs. Physical decomposition – Example (2/4)



SysML challenge: Logical vs. Physical decomposition – Example (3/4)



SysML challenge: Logical vs. Physical decomposition – Characteristics



What to do

- Model and trace requirements, functional design, logical design and buildable system without separation of logical and buildable design to reduce modelling overhead.

How to do

- Describe functional design with activity diagrams and state machines
- Use logical blocks to describe main architecture and is interfaces
- Allocate functions to logical blocks
- Use logical blocks as context for concrete blocks
- Re-Allocate functions to concrete blocks
- Allocate blocks if connectors are irrelevant or still unknown and configuration changes easily, i.e. blocks are loosely coupled.

Summary

What we have tried

- Check the usability of SysML for space observation domain
- Reach the limits of SysML for system engineering of
  - Requirements
  - Structure
  - Behavior

Our current conclusion

- SysML can be used to model space observation systems
- SysML offers not much built-in opto-electronical engineering
- We have reach some limits of SysML
- However: Do not use to much fancy SysML constructs
  - Common understanding of all system engineering stakeholders is the most important value