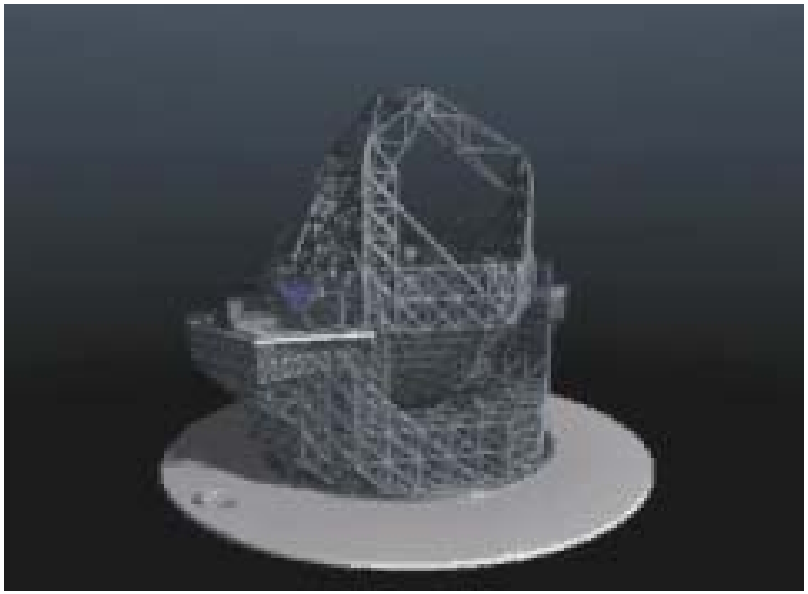


# SysML for Telescope System Modeling Proceeding II – 2009-04-15

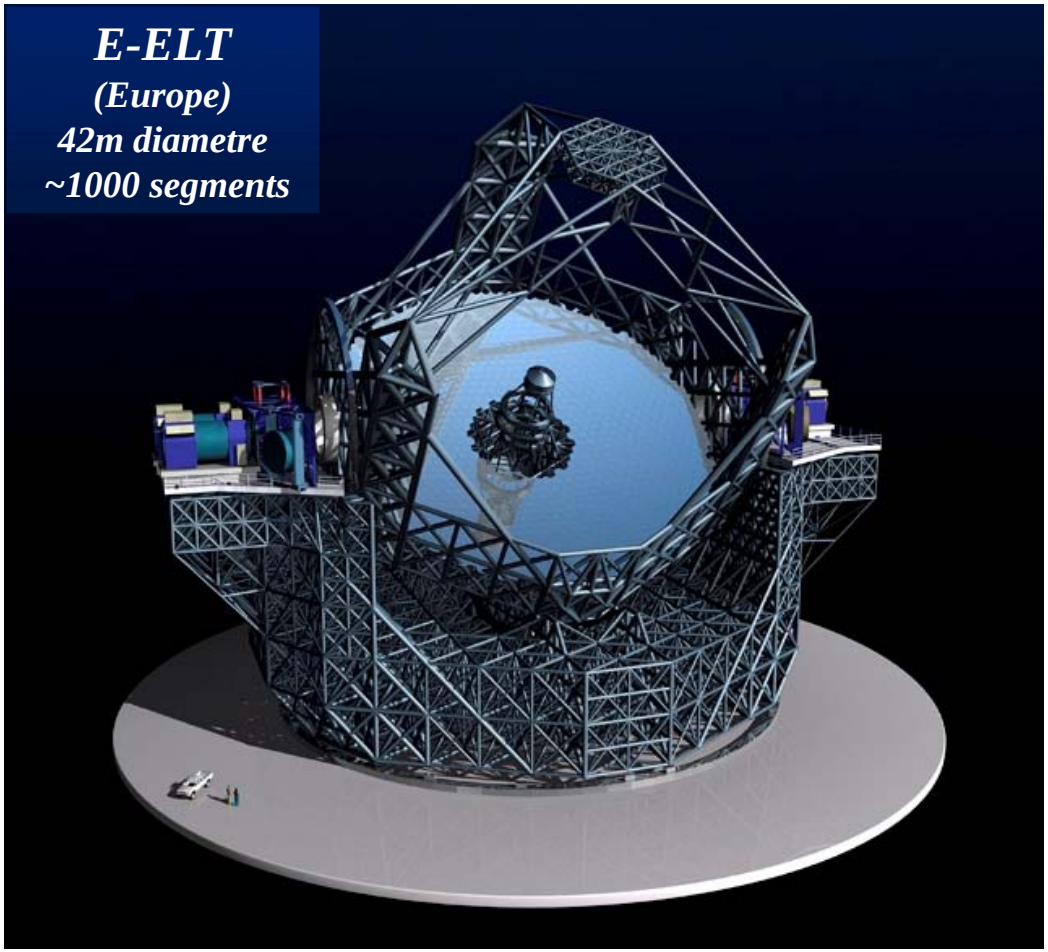
**INCOSE MBSE Challenge Team SE^2**
**Robert Karban (ESO)**
**Tim Weilkiens (oose GmbH)**
**Rudolf Hauber (HOOD Group)**
**Andreas Peukert (TUM)**

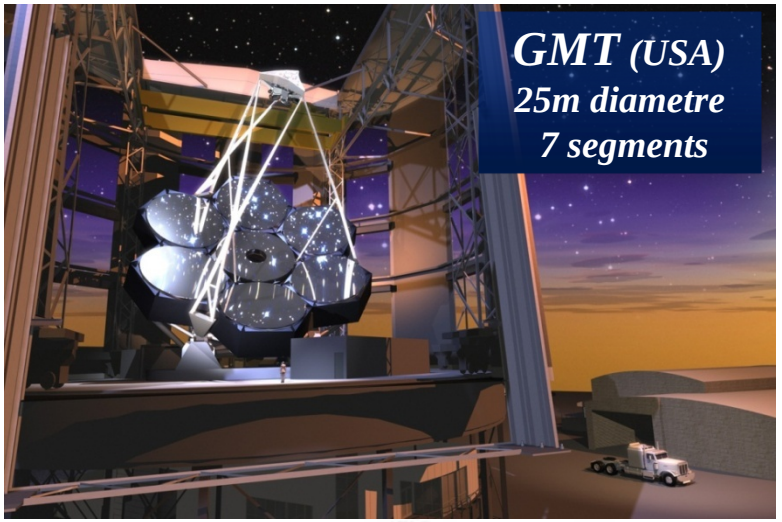
*Images on this slide were produced by ESO*

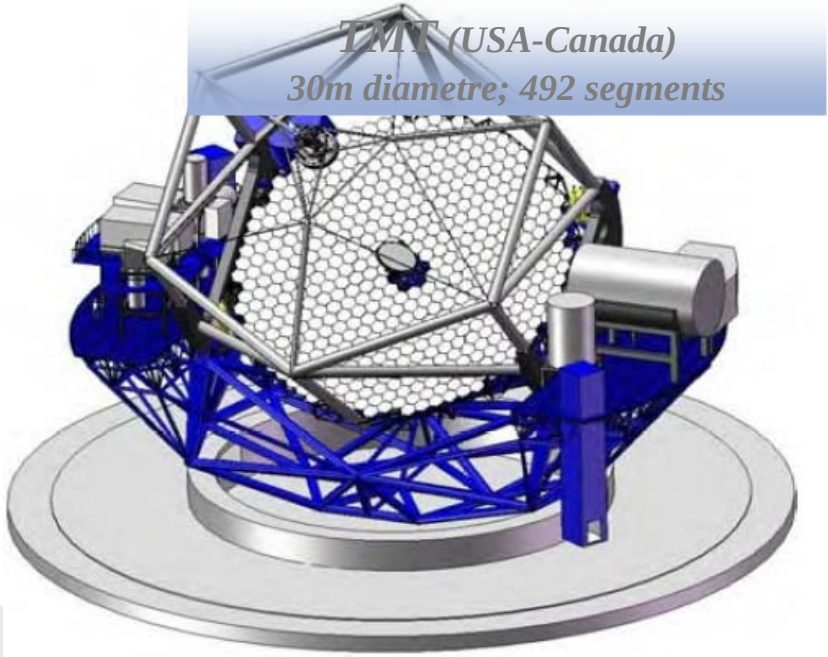
## Agenda

- What is the Challenge project about?
- The goals of the SE^2 team
- Reminder of some of the results
- Status of Issues identified at IS09

# MBSE Challenge Team SE^2
# SysML for Telescope System Modeling

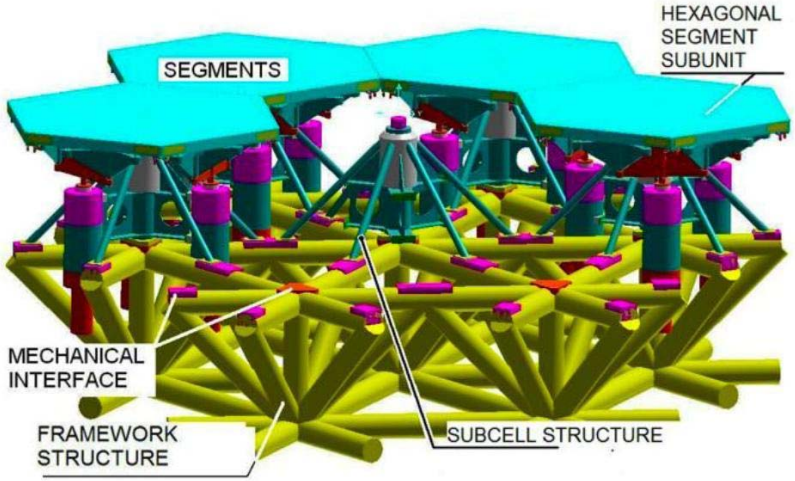


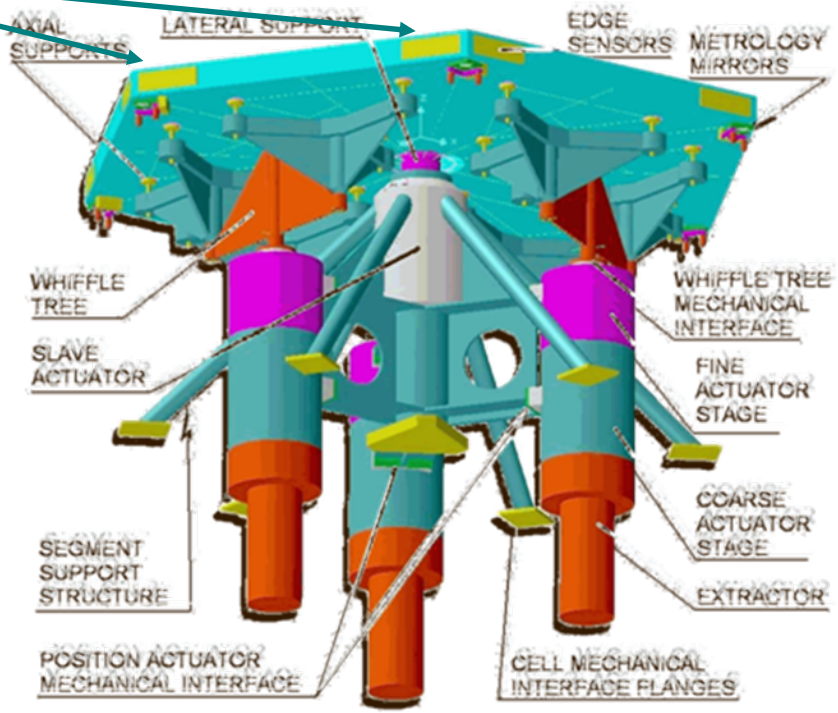



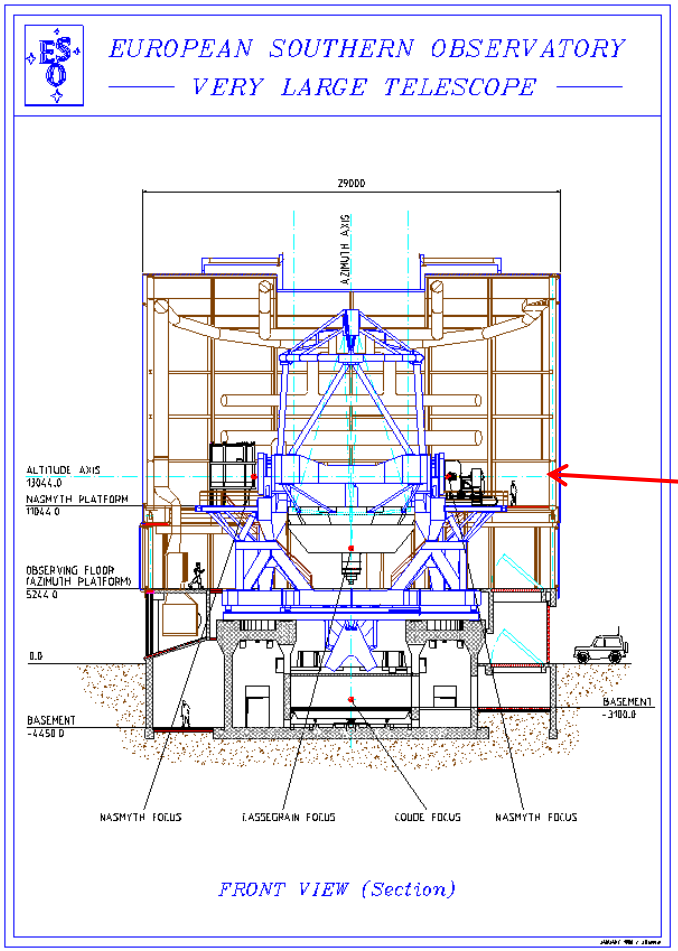

Presented at the INCOSE MBSE On-Line Summit April, 15th 2009

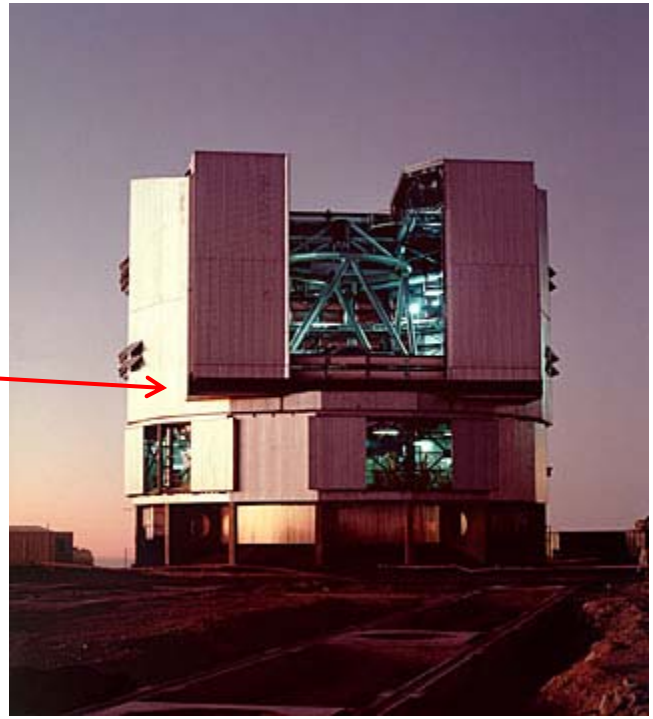
# Edge Sensors



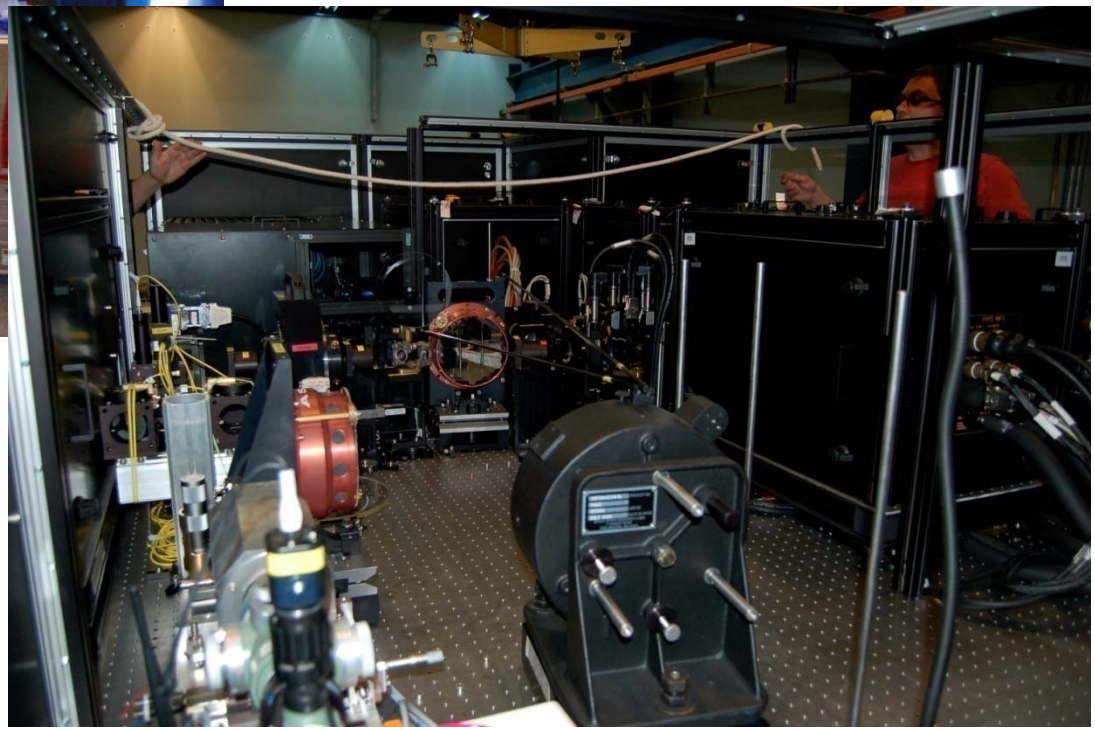

Detect nanometers of phasing error in micrometers of turbulence with Phasing Wave Front Sensors (~20 nm RMS)



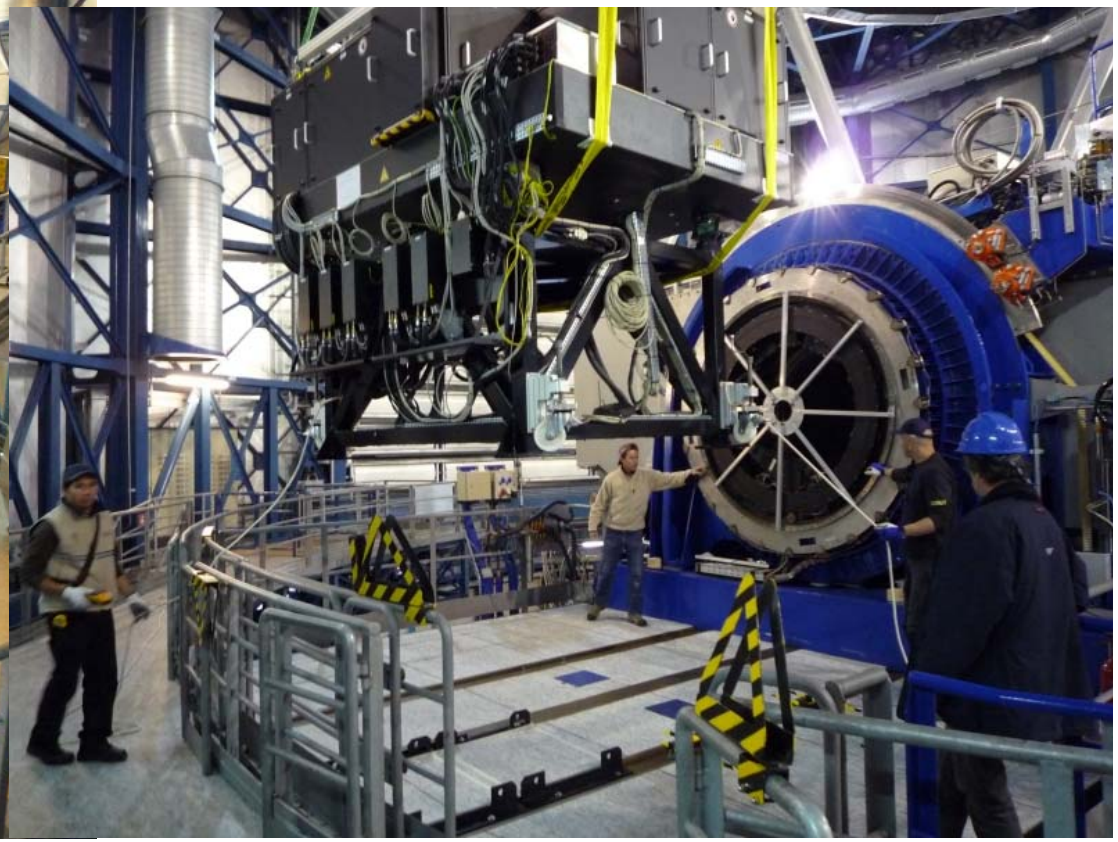



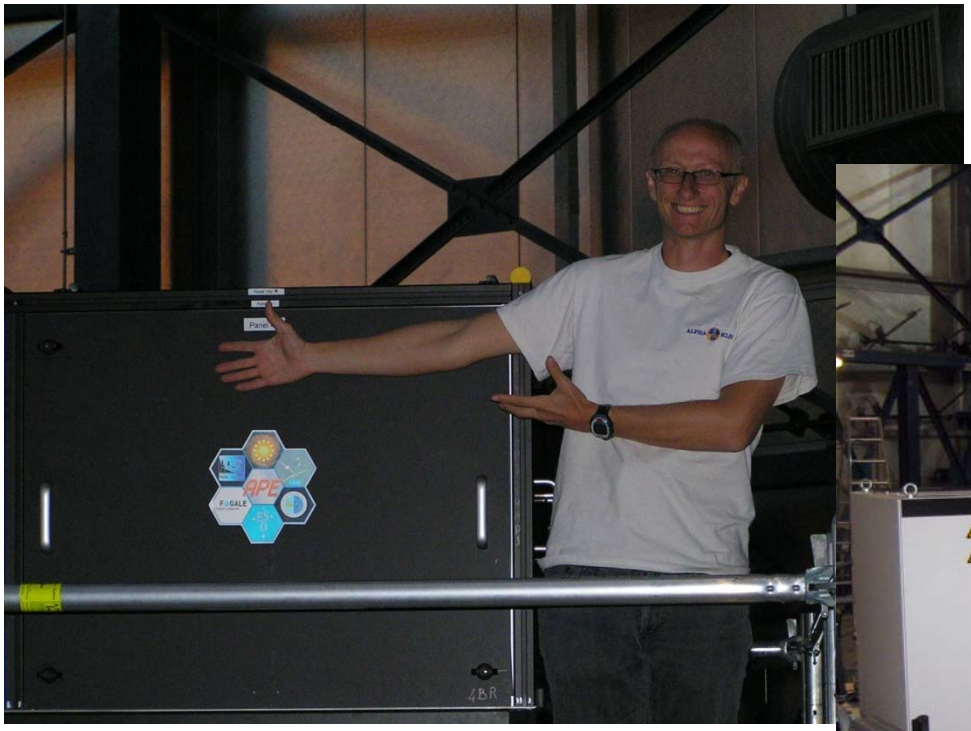
*APE will be installed at the telescope in the Chile desert.*

# Assembling the pieces in the integration hall



# Installation on the platform of the telescope





SysML in practice ;-)

## 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?

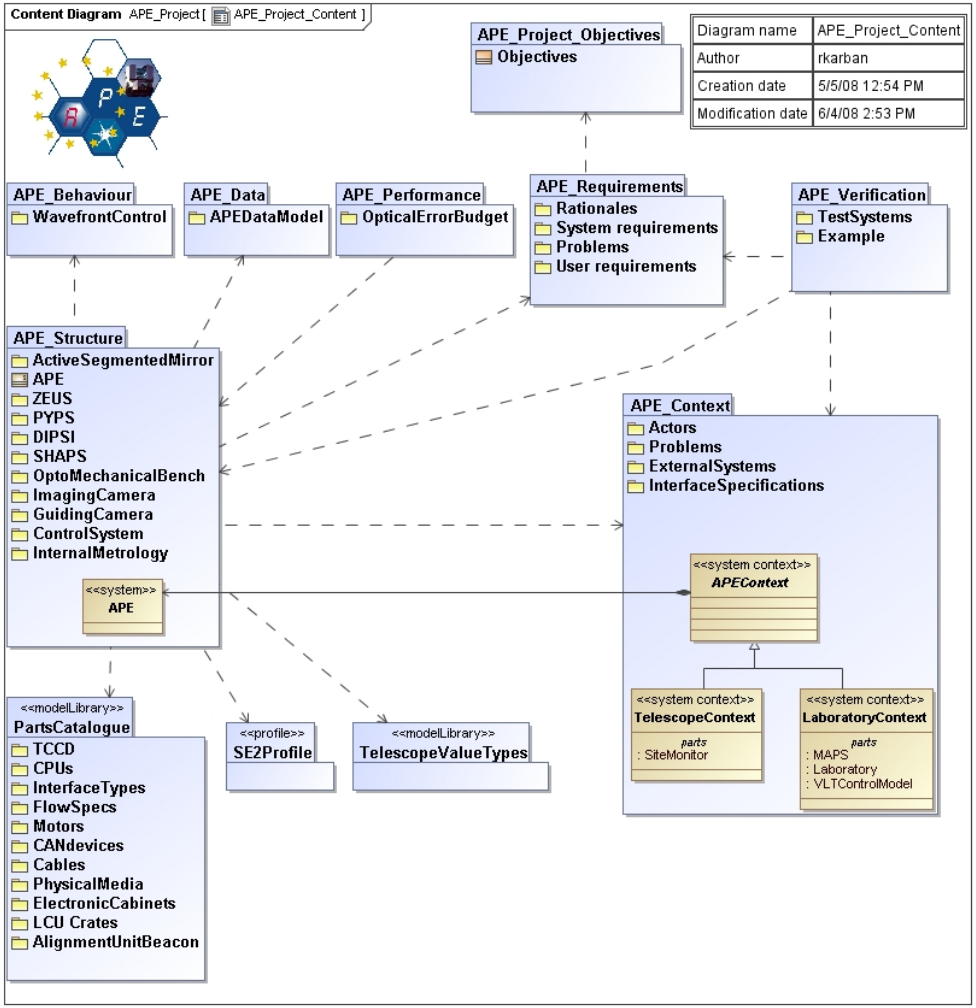
## Deliverables:
## Generic SysML modelling FAQ: Excerpt 2/2

- 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?

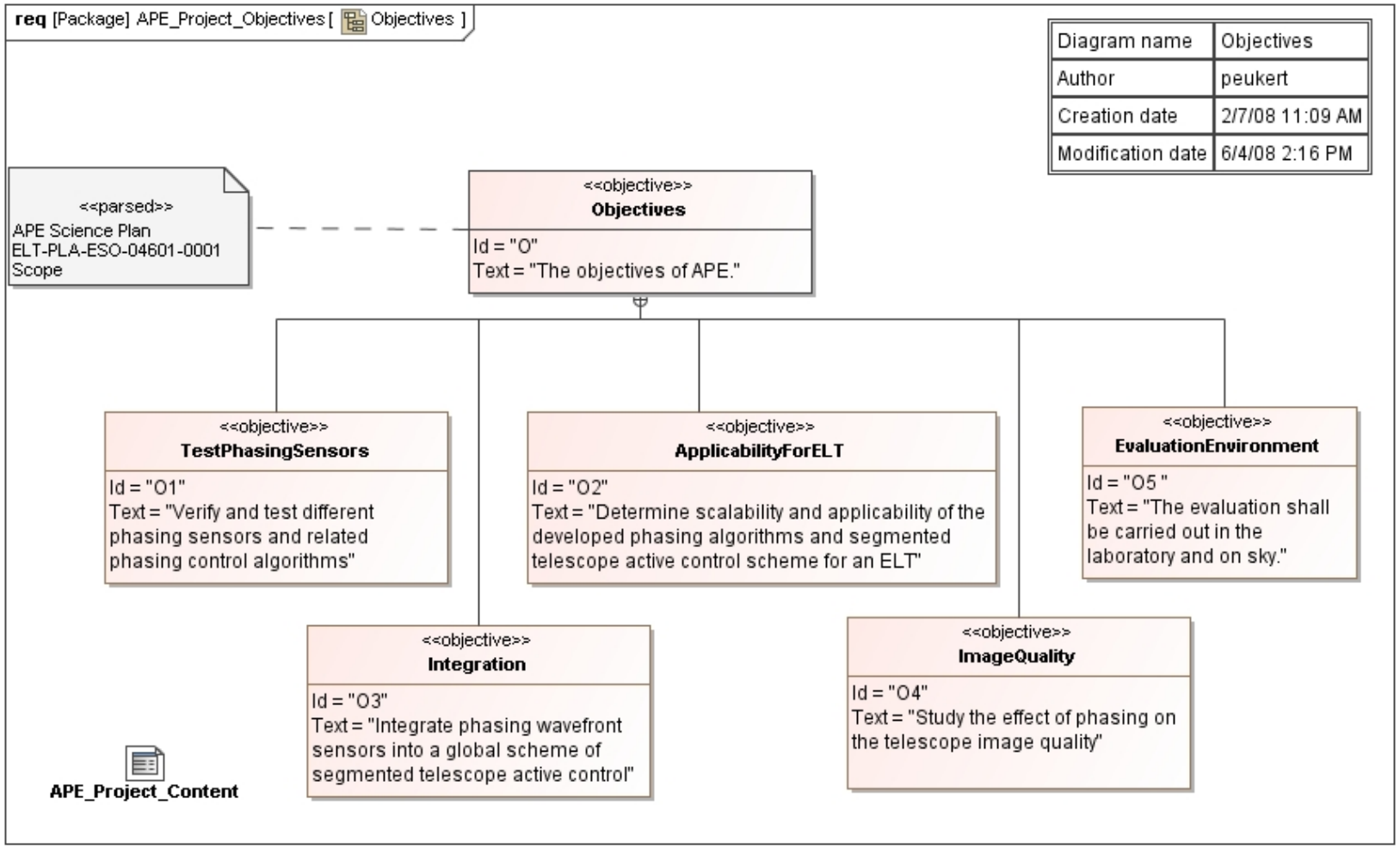
## Deliverables:
## SysML model for the APE project

- 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
- **Documented at: http://mbse.gfse.de**

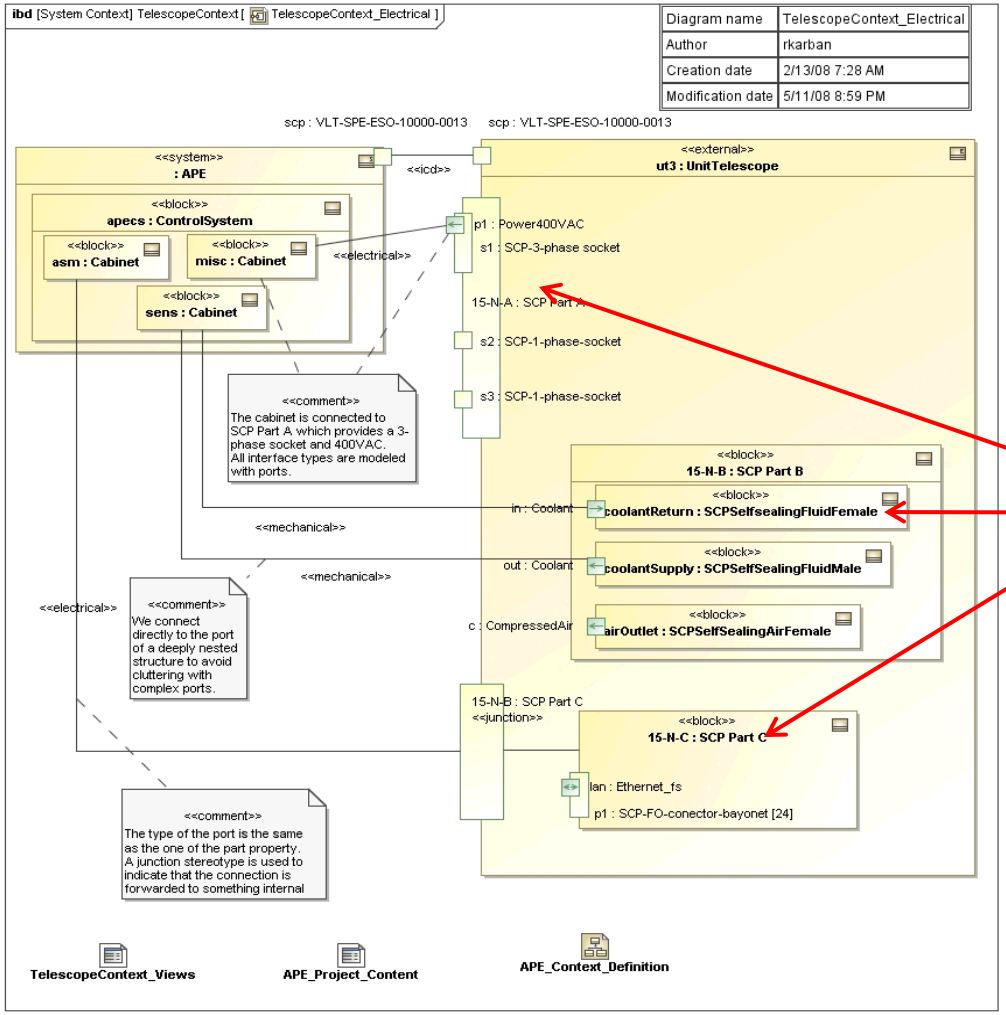
# APE project : Overview



# APE project: Objectives / requirements



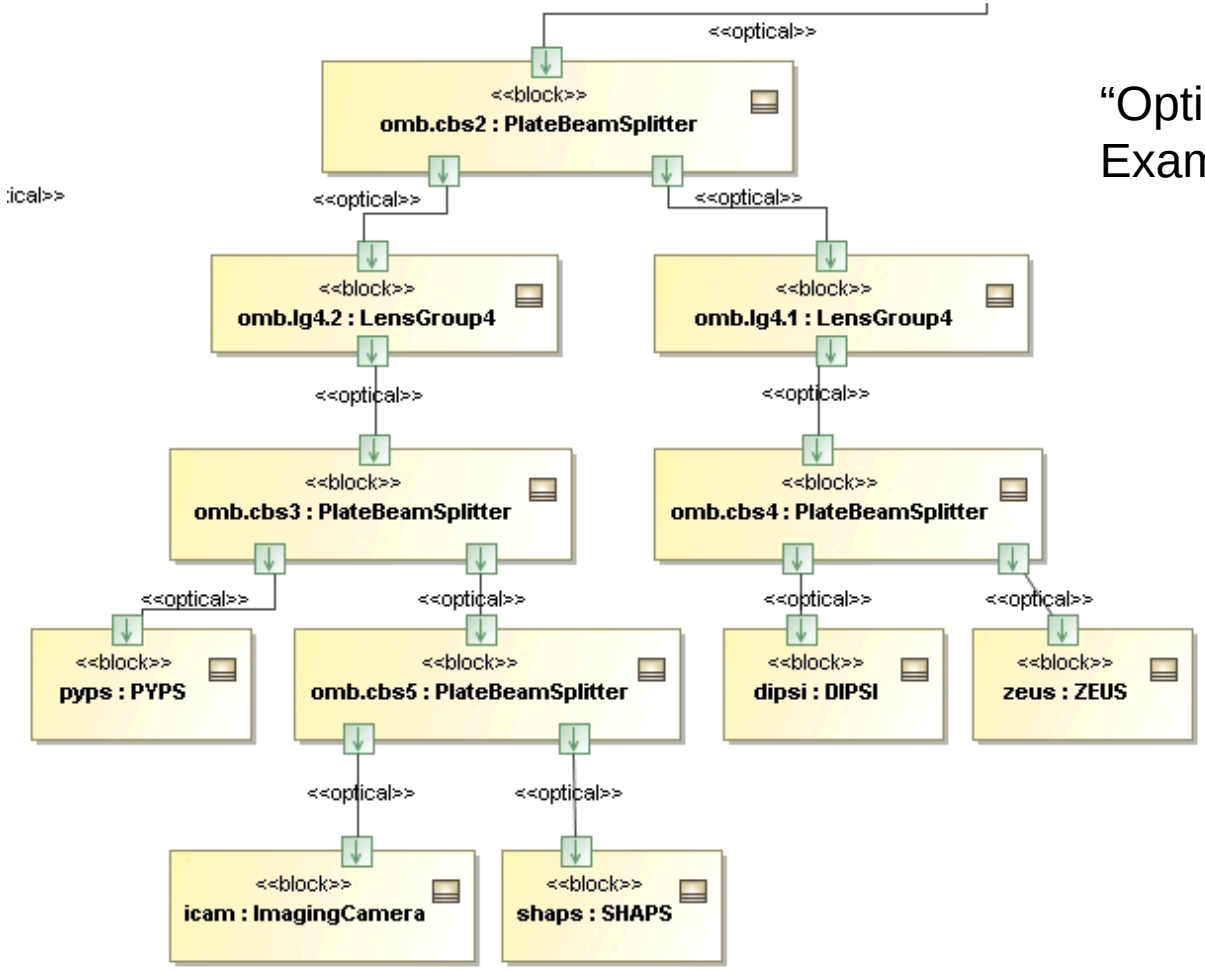
# APE system model: System context




3 modeling approaches for interfaces

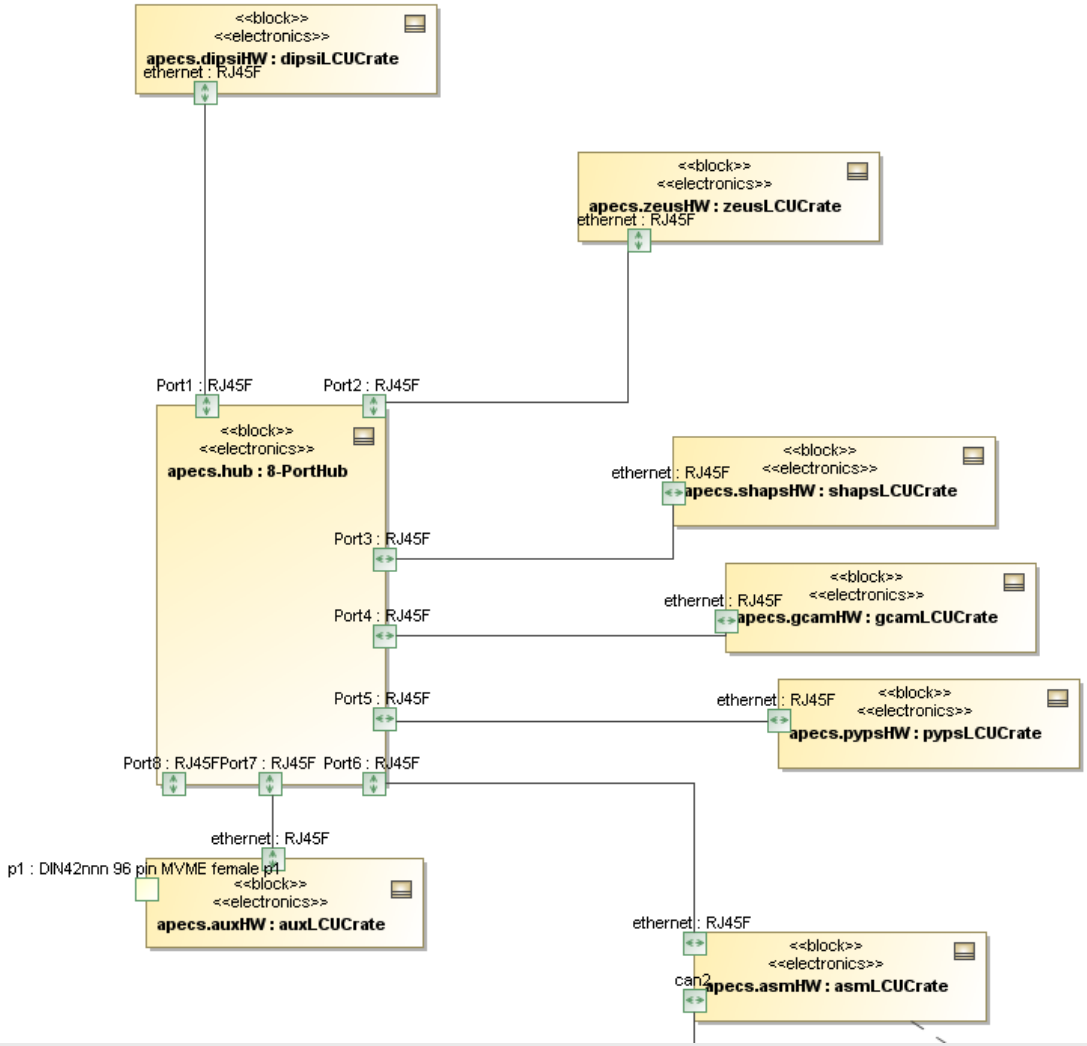
➔ treated later in challenges

## APE system model: Structure: Internal structure




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

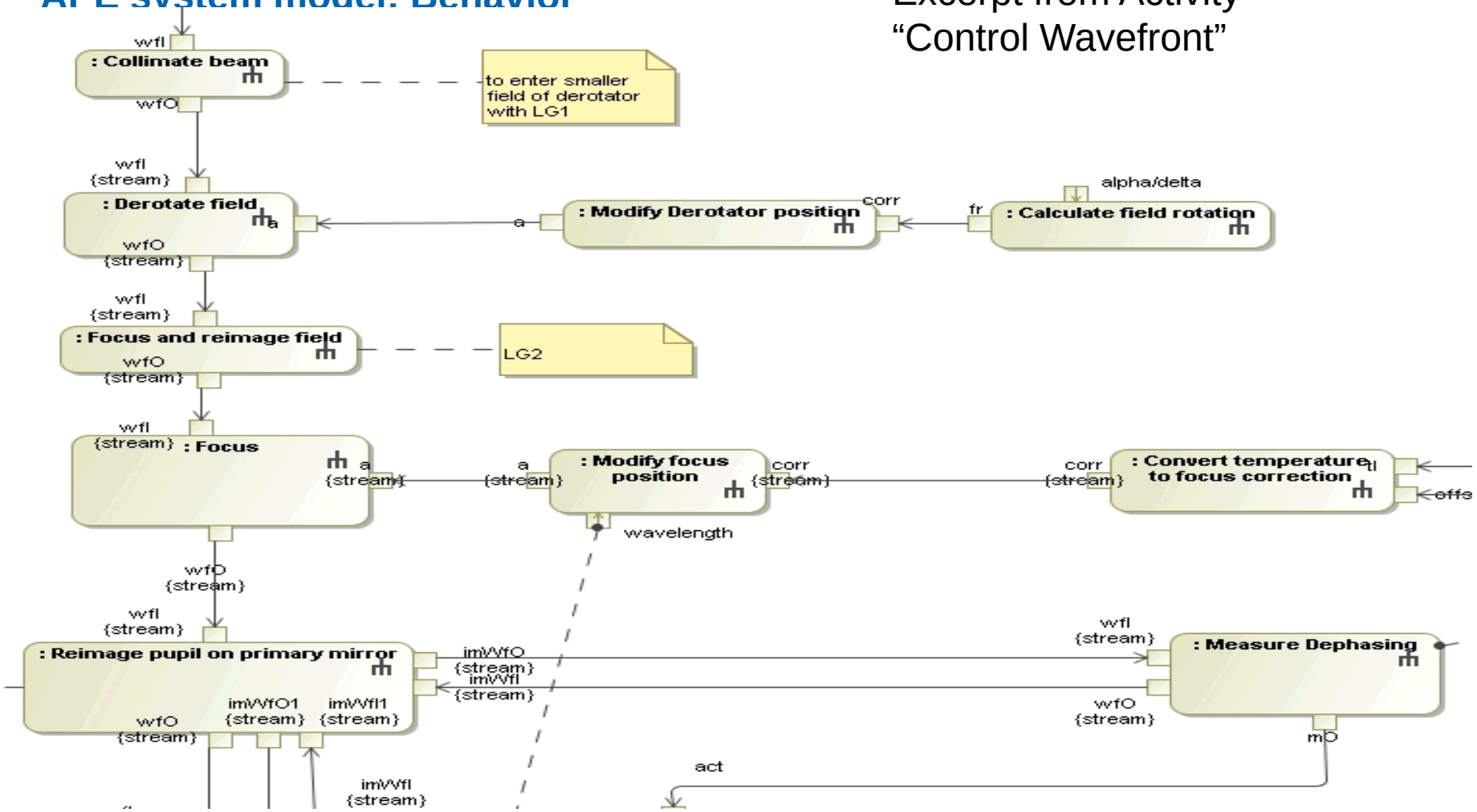
# APE system model: Structure: Internal structure




"Electrical view" of APE

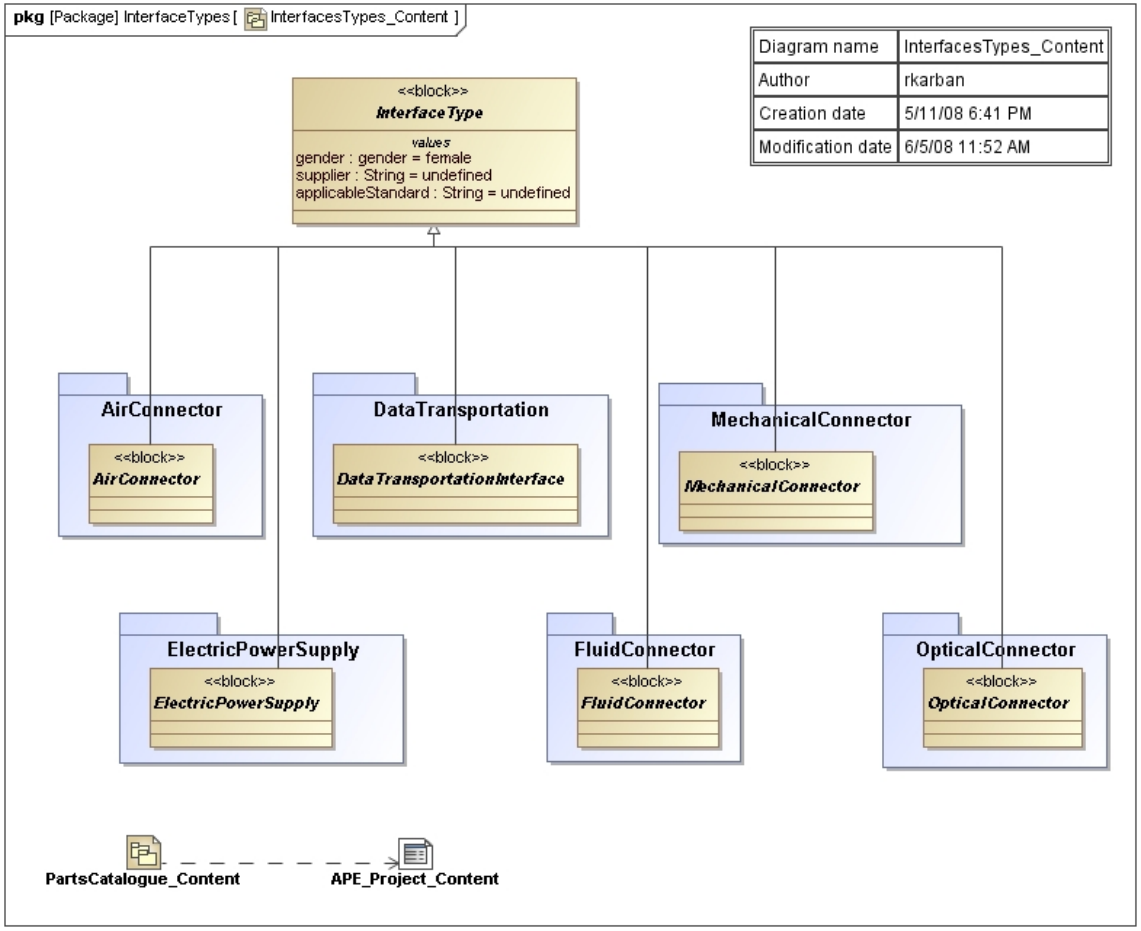
## APE system model: Behavior

Excerpt from Activity
“Control Wavefront”



## Catalogue model: Abstract types

Example for catalogue:

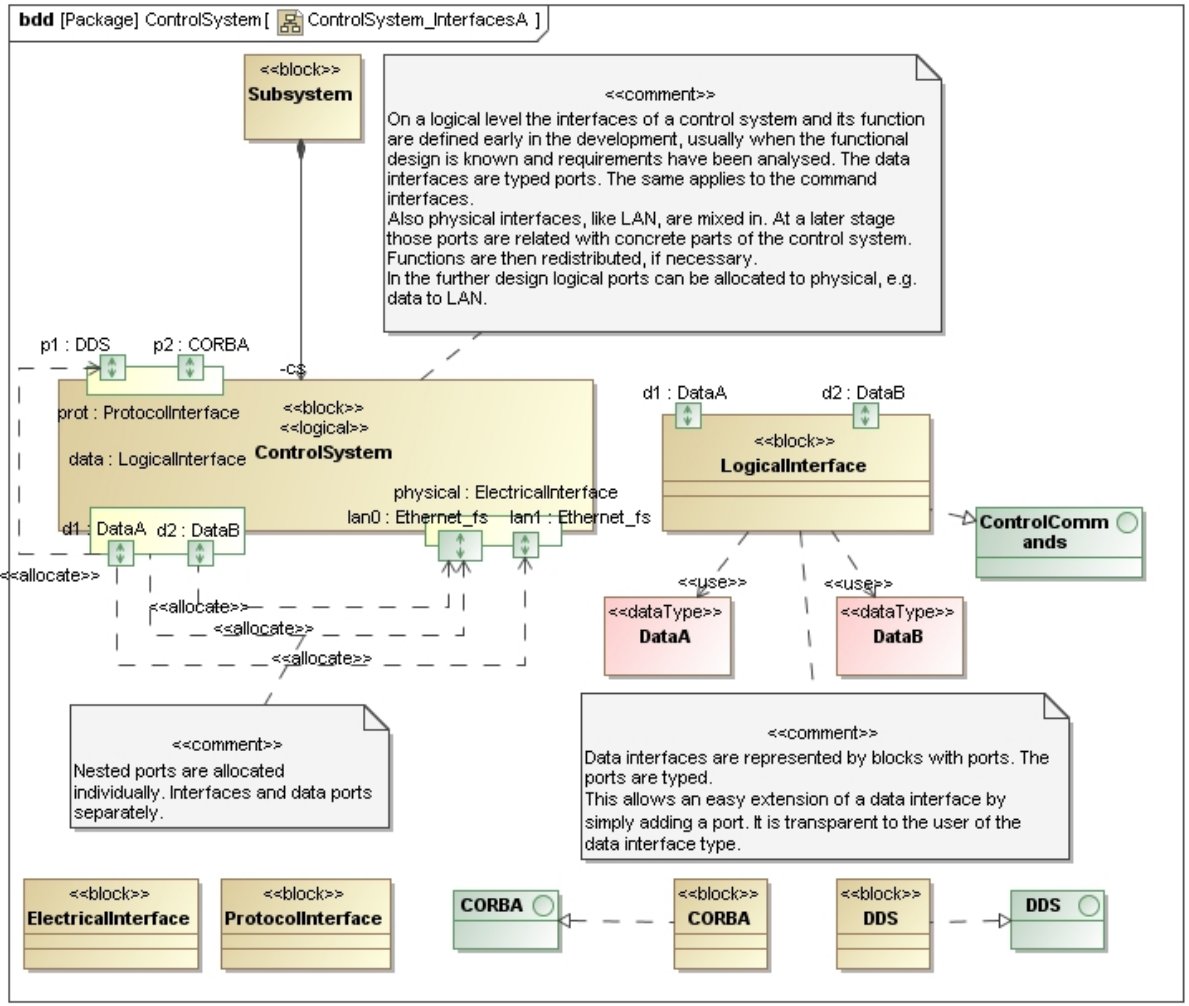


## Challenges identified at IS08

- *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 allocation**
- **Defining Quality of Service (QoS)**
- **Transition to UML for software**
- **Configuration and Quality Control**
- **Navigability**
- **Deployment in an organization**
- **“Instance values”**

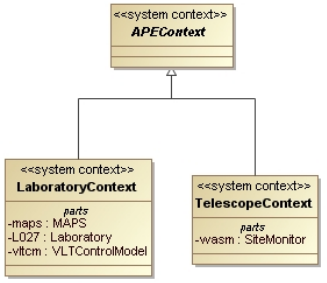
Note: *Order has no meaning, e.g. priority*

# SysML challenge: Structural multi-layer allocation – Example (1/1)




| | +lan0 : FlowSpec... | +lan1 : FlowSpec... | +d1 : Playground... | +d2 : Playground... | +p1 : Playground... | +p2 : Playground... |
|---|---|---|---|---|---|---|
| ControlSystem | 1 | 1 | 2 | 1 | 1 | |
| ProtocolInterface | | | 1 | | | |
| +p1 : Playground::Inte... | | | ↙ | | | |
| +p2 : Playground::Inte... | | | | | | |
| LogicalInterface | 1 | 1 | | | 1 | |
| +d1 : Playground::Inte... | | ↗ | | | ↗ | |
| +d2 : Playground::Inte... | ↗ | | | | | |
| ElectricalInterface | | | 1 | 1 | | |
| +lan0 : FlowSpecs::Eth... | | | | ↙ | | |
| +lan1 : FlowSpecs::Eth... | | | ↙ | | | |

## SysML challenge: Structural multi-layer allocation – Characteristics




### Notion

- Different logical interfaces can use different physical interfaces, e.g LAN port, or protocols, e.g. CORBA

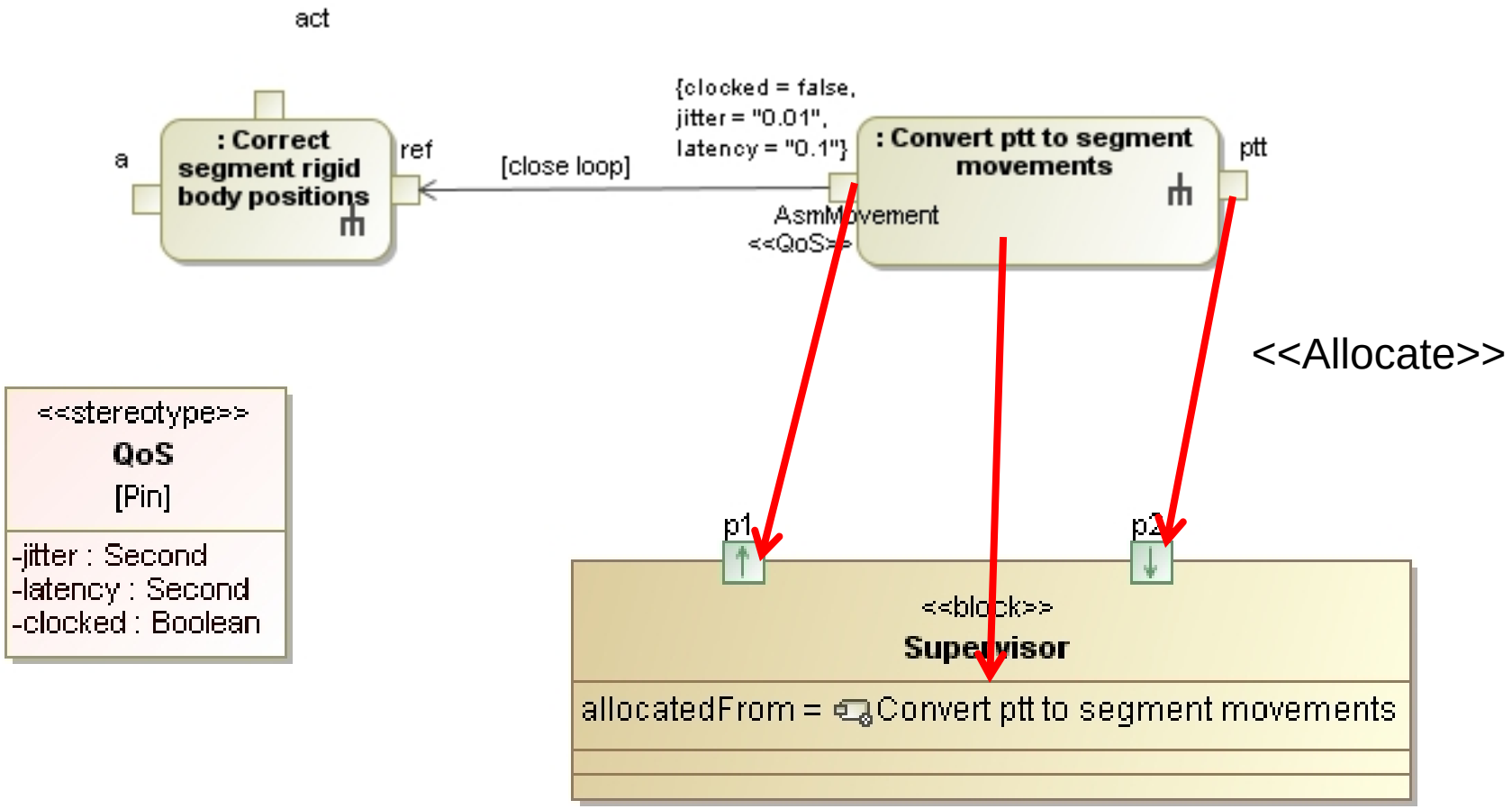
### How to

- <<Allocate>> logical ports to physical ports and protocol ports
- Profile with stereotypes for interfaces types
- Special port types for better readability (cluttered diagram by stereotypes)

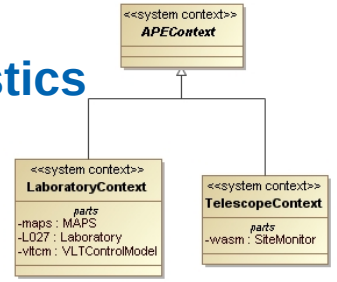
### SysML status

- There are no plans to support discipline specific interfaces types. That would be contradictory to the unified approach of SysML. It is a task for the stereotypes mechanism.
- Allocation is a stereotype of UML abstraction and the semantics (i.e. the exact mapping) of allocate are not defined in SysML. Mapping to be defined. For practical reasons use a Note.

# SysML challenge: Defining Quality of Service (QoS) – Example (1/1)



# SysML challenge: Defining Quality of Service (QoS) – Characteristics




## Notion

- Additional information on object flow for modelling of performance details, like jitter, latency, etc.

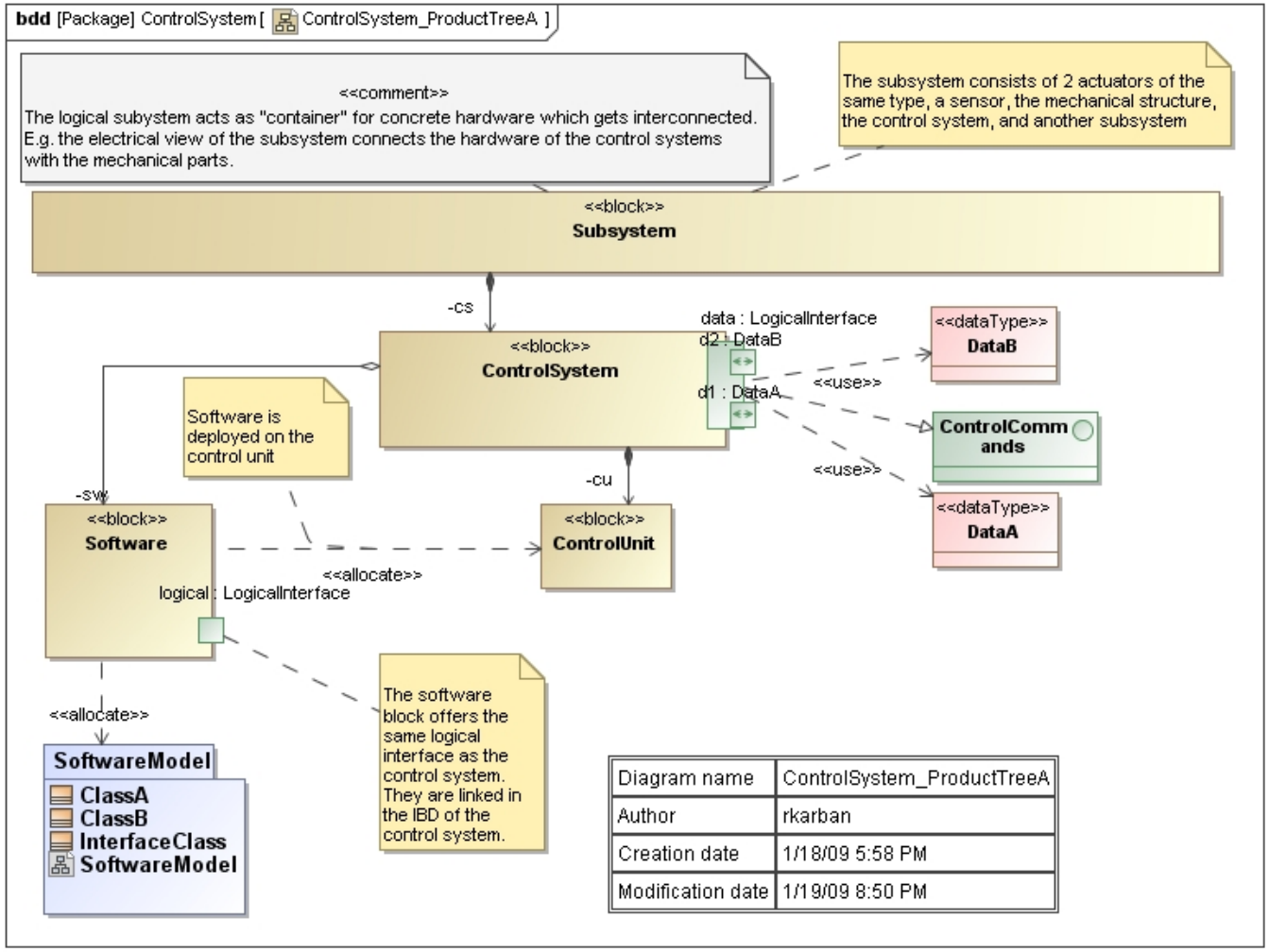
## How to

- Create <<QoS>> stereotype(s) for pins/parameters and fill in tags for each
- Allocate activity and pins to blocks and ports, if necessary
- Disable Actions in tool

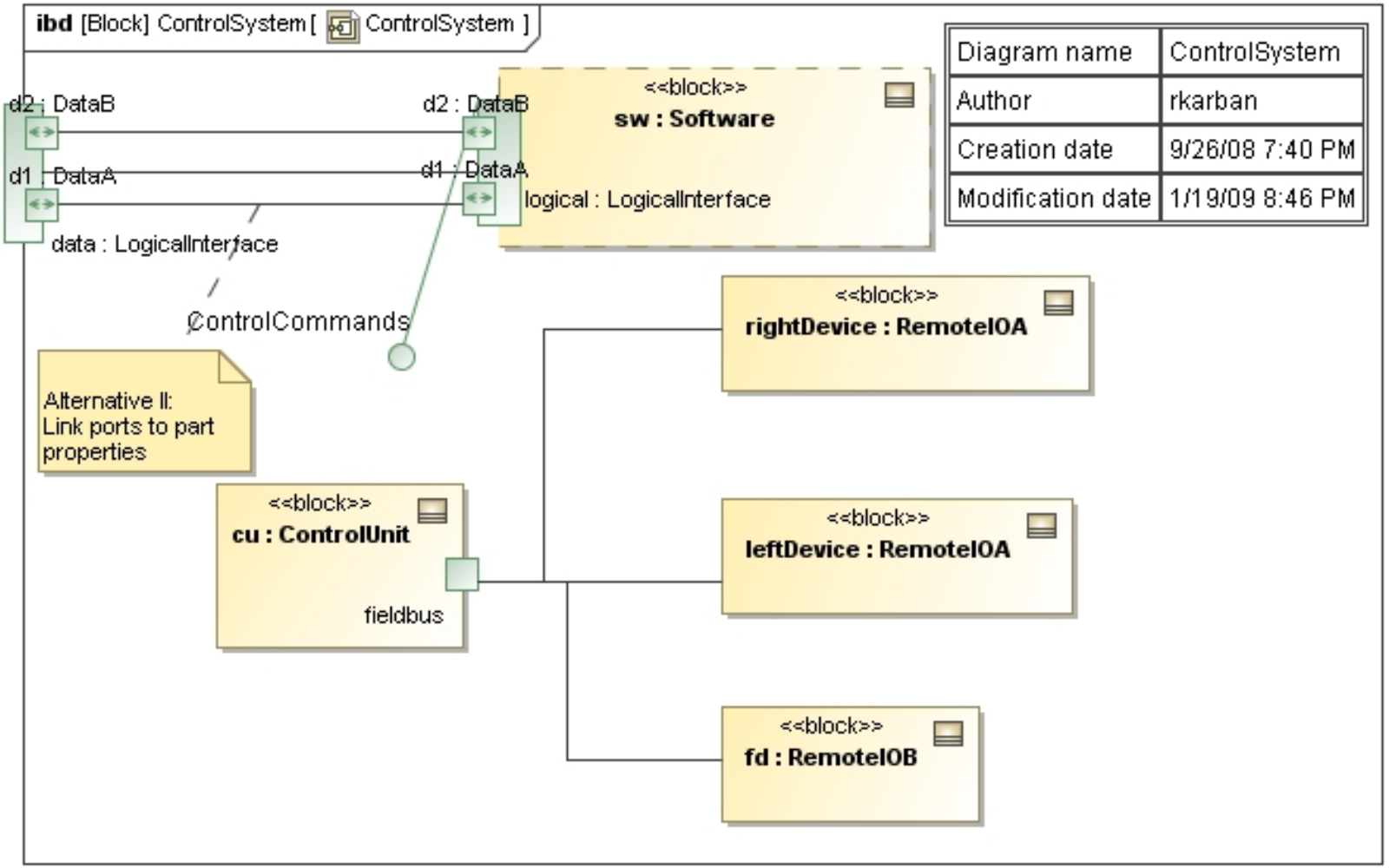
## SysML status

- SysML only provides only <<rate>> stereoype which extends Activity Edge and Parameter.
- Allocation of Ports to Pins not addressed in SysML standard 1.1
- MD extends these stereotypes to ObjectNode for applications like this.
- Synchronization of Parameter and Pin is tool-dependent.
- UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms

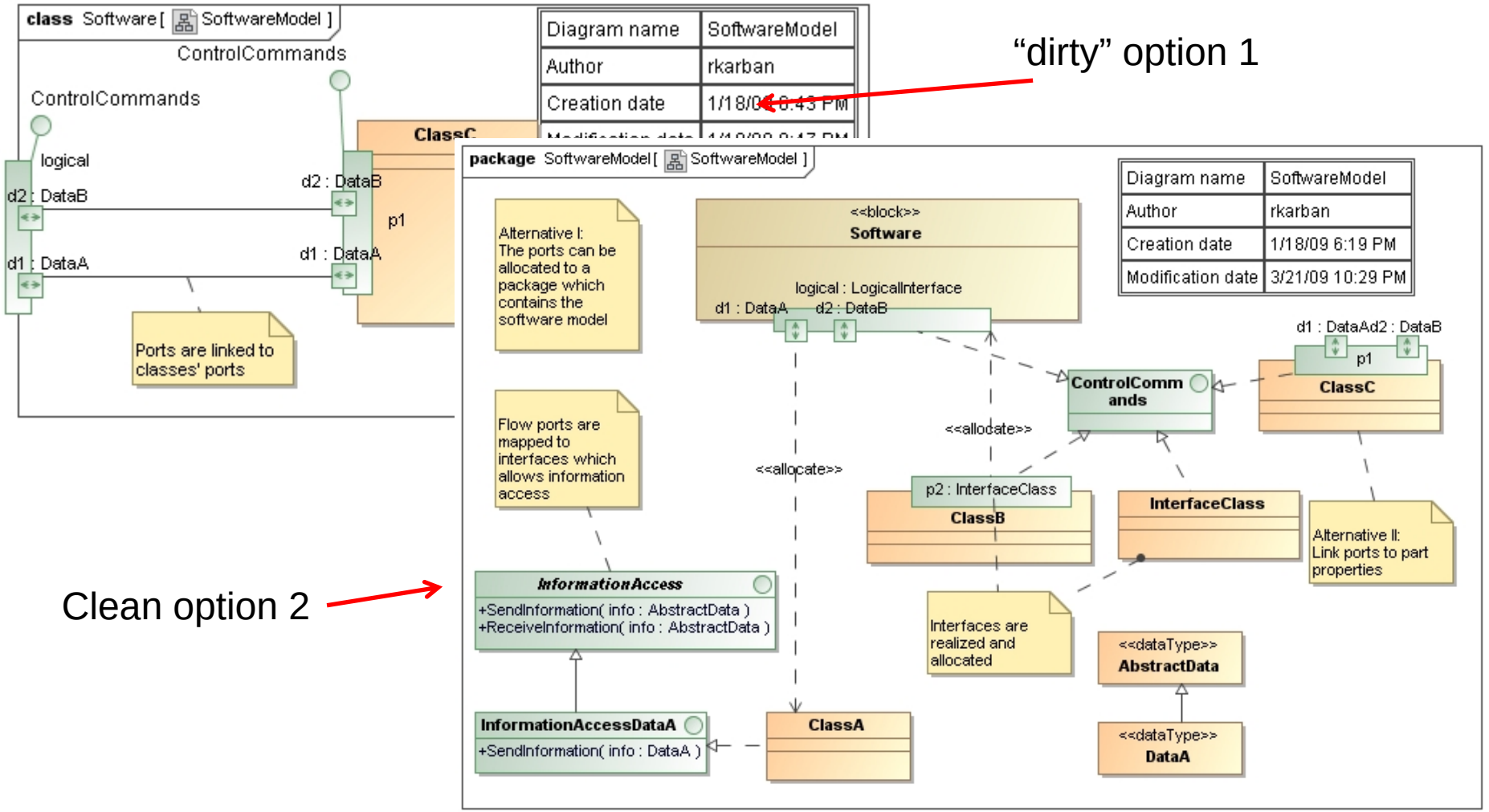
# SysML challenge: Transition to UML for software – Example (1/3)



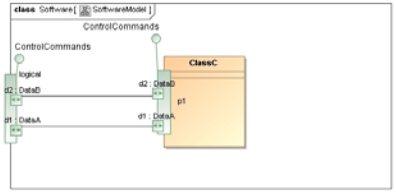
# SysML challenge: Transition to UML for software – Example (2/3)



# SysML challenge: Transition to UML for software – Example (3/3)



## SysML challenge: Transition to UML for software – Characteristics




### Notion

- Seamless transitions from SysML <<system>> and <<software>> blocks to UML classes, mapping also ports and interfaces

### How to

- <<allocate>> block to package (a la M. Hause)
- Alternative I: <<allocate>> SysML ports to UML ports and <<realize>> the same interfaces. Use interfaces for information access to map flow ports.
- Alternative II: create a UML „part class“ representing the SysML block and create connectors for SysML ports to UML ports in IBD and class diagram

### SysML status

- <<allocation>> implies dependency of System to SW or vice versa.
- Classes are excluded from SysML

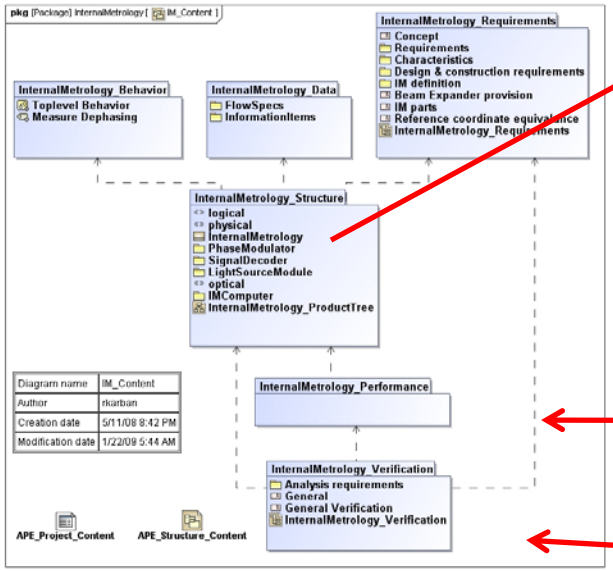
## SysML challenge: Configuration and Quality Assurance

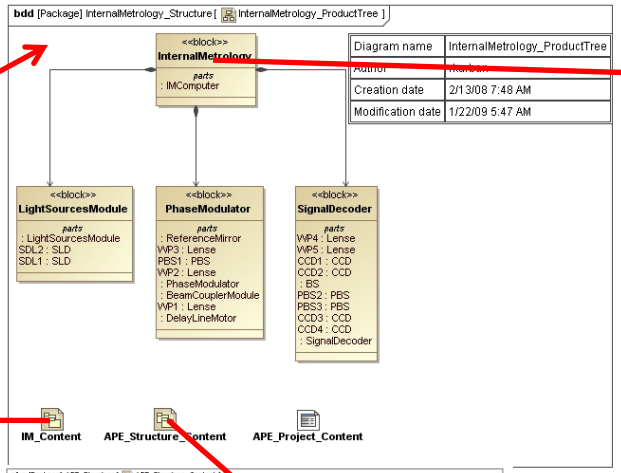
- Require coordinated configuration control of modelers
- Require to know which parts have changed – changes can happen anywhere in the model
- Require Track Changes a la Word for individual commit, visual diff

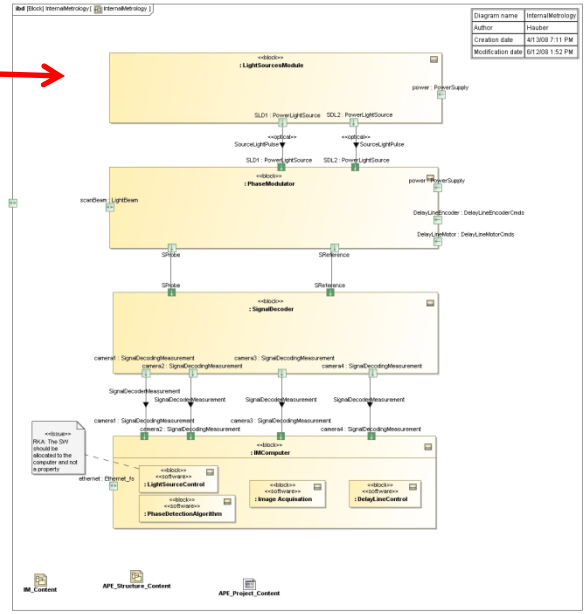
- MagicDraw Team Work Server (TWS) solves some
  - Locking at model element/symbol/diagram level by different users
  - No commit of individual parts only of complete project -> unknown which parts changed (visual diff), changelog related to complete model
  - Problem in the beginning with corrupted IBDs -> fixed very quickly
  - Allows working offline/local with deferred merge

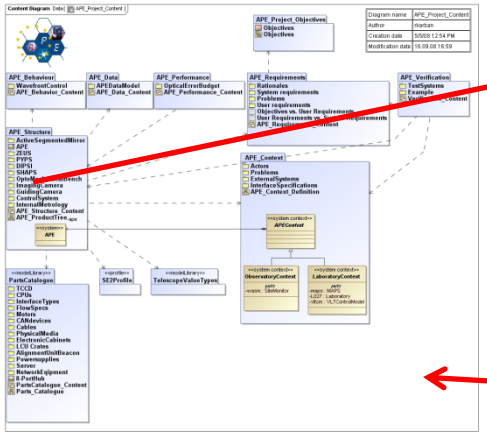
- Provide template for structure (getTemplate Plugin)

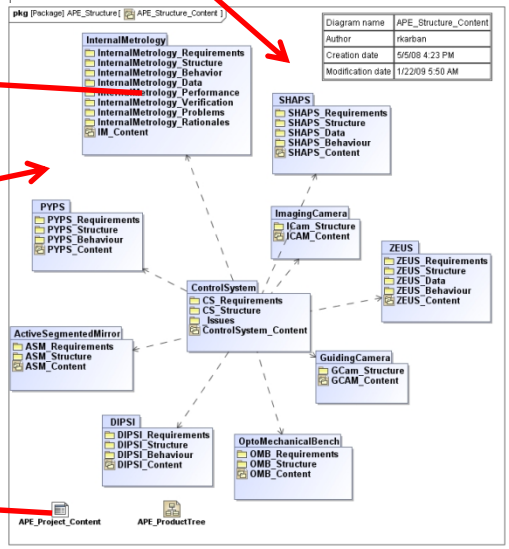
# SysML challenge: Navigability – Hyperlinks - Example (1/2)



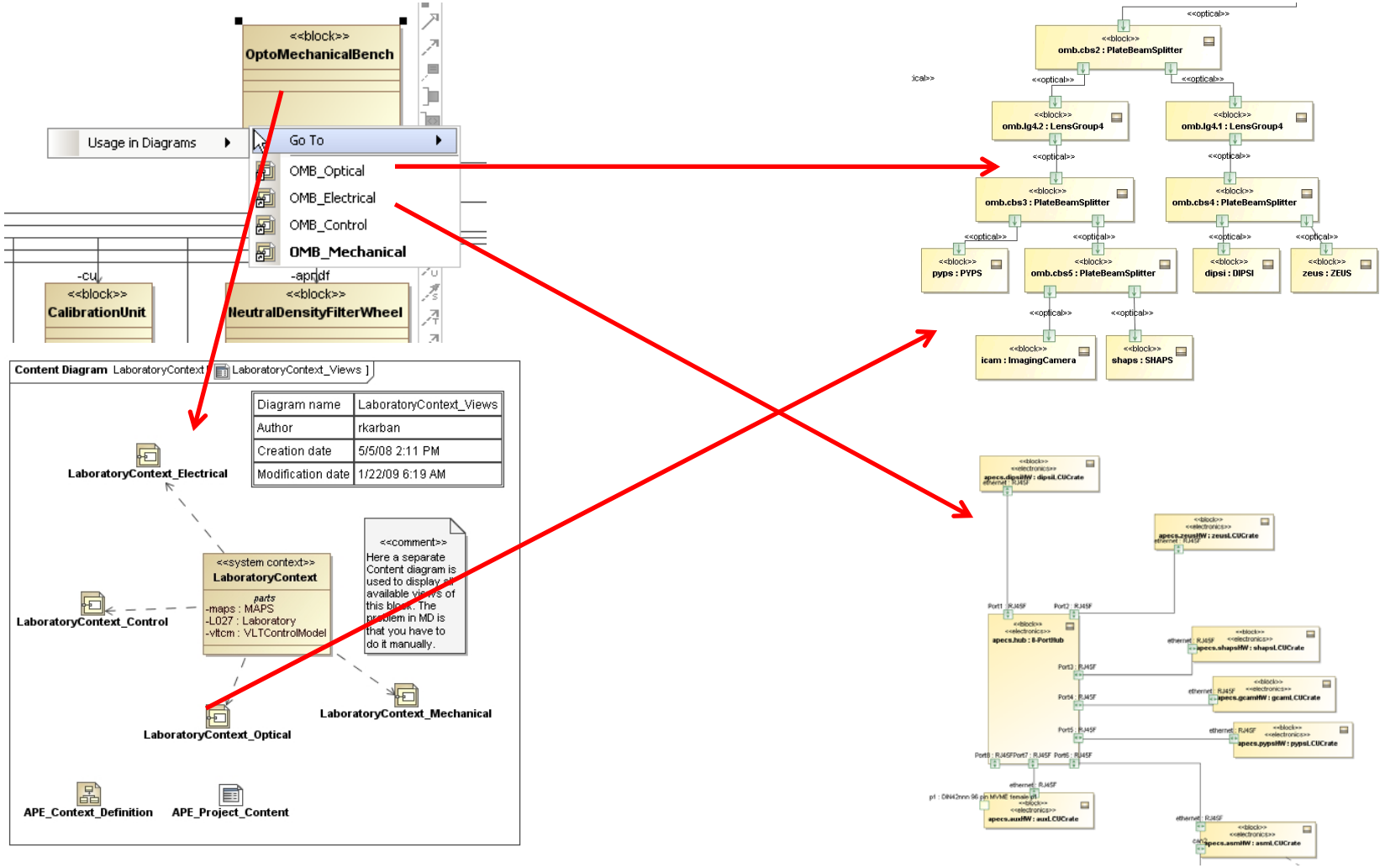



# SysML challenge: Navigability – Multiple Views - Example (2/2)



## SysML challenge: Navigability of models – finding the things you need

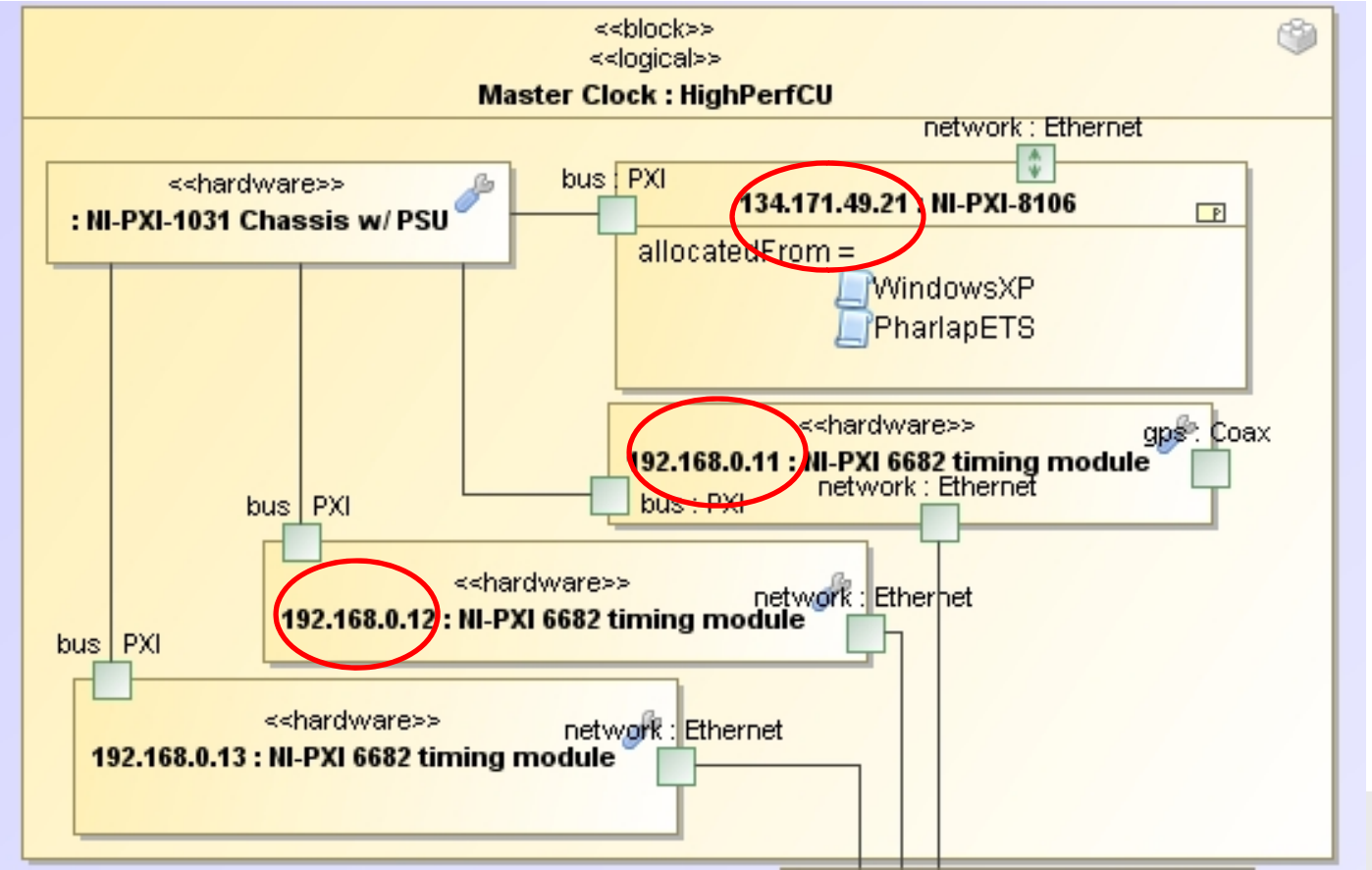
- Basic Rules
  - At creation of an element: *"What can I hyperlink it to ?"*
  - Assembly Block to its Internal Block Diagrams (IBD) – multiple views
  - Single model or package to a SysML Package Diagram (or SysML Block Definition Diagram)
  - Part Property to its Internal Block Diagram (IBD)
  - From every diagram to top level diagram
- Navigate on elements and packages – only little the browser to „OPEN UP“ things
- Hyperlink packages with contents list and dependencies between packages to reflect process

## Challenges of SysML deployment in an organization

- Best practices
  - Mentor and SysML/Tool confident person
  - Extend gradually the range, define modeling goals, guidelines, and standards
  - “Just use it!” (Do not talk about modeling and SysML too much as it raises fear of waste of time)
- Observations
  - (no) support/commitment from management  but a necessity for engineering
  - How is presented to management? How do they see a gain?
    There is no immediate real-life artifact (no LED blinking, no tangible objects)
  - Under pressure people fall back to techniques they know
  - People are often lazy to learn/apply something new
  - Not modeling means often not understanding and therefore underestimating the problem.
  - Modeling reveals complexity and people get scared
  - Contractual problems with models – only text is understood by lawyers

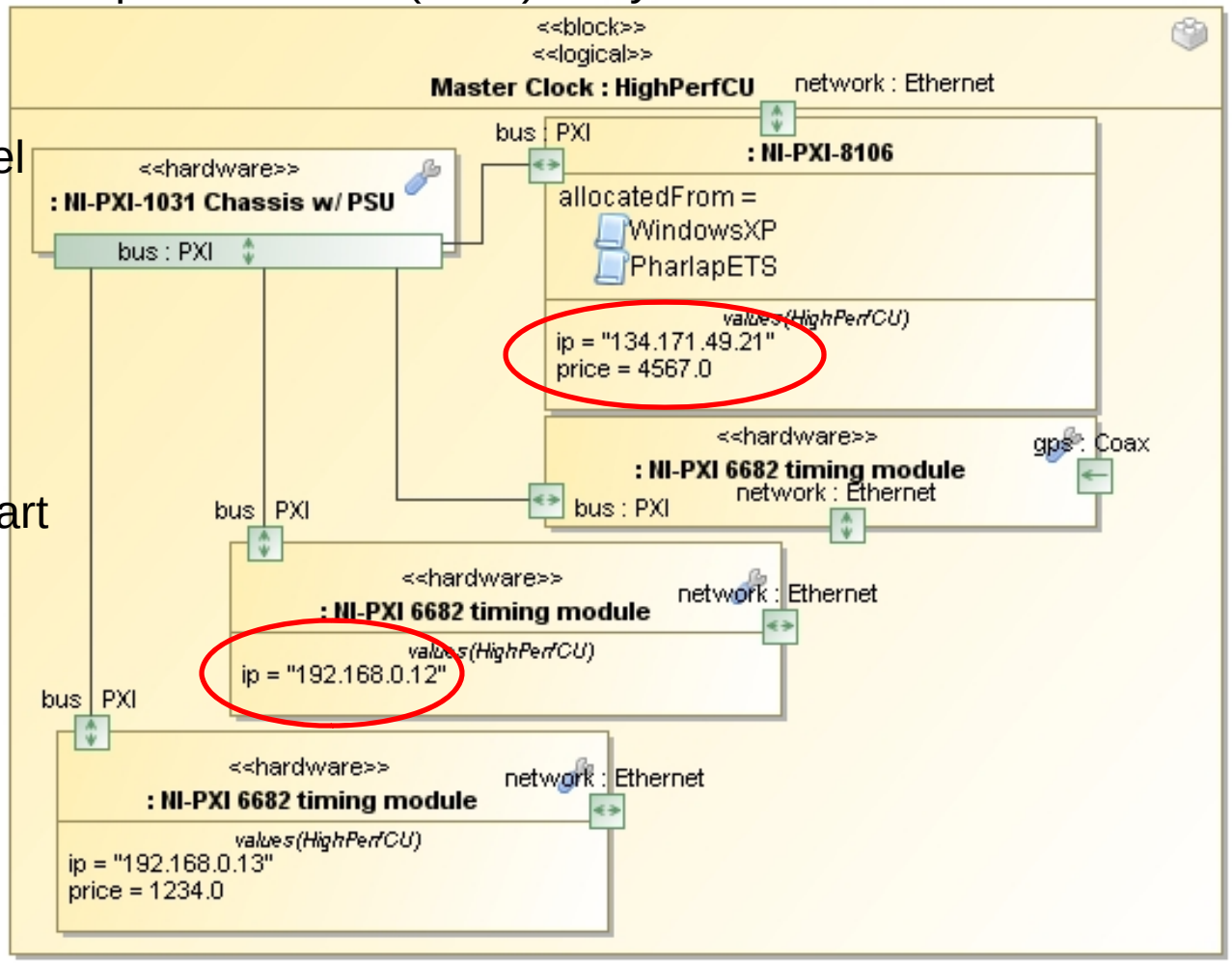
## Challenges of SysML: “Values” – with work around 1/3

- Unclear definition of context specific values to value properties of part properties causes weird workarounds in SysML 1.0 for IBDs.
  - E.g. defining the IP address for a part-property of a block representing a PC



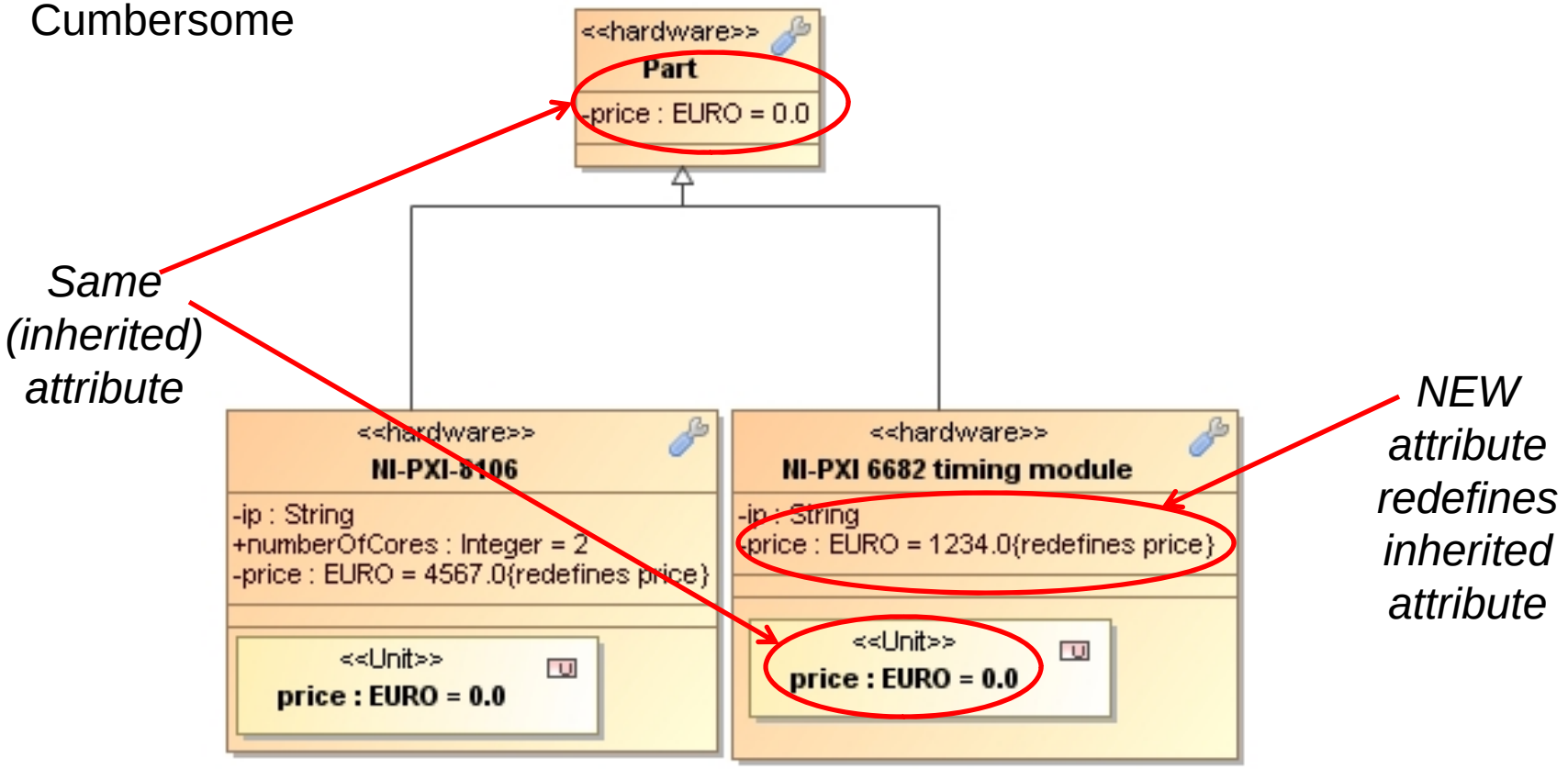
## Challenges of SysML: "Values" – Context Specific Values 2/3

- Better definition of context specific values (IBDs) in SysML 1.1 to define values at usage level!
- Notation defined but not mapping into model (tool dependent)
- Owner can only be a package -> difficult to relate to part
- InstanceSpec should be nested within the part



## Challenges of SysML: "Values" with Property Specific Types 3/3

- Property Specific Types are used to define values at Block definition level (default value) – could use Instances but don't know how to relate to structure
- Cumbersome




.

## Summary

### What we have

- Checked the usability of SysML for ground based astronomy domain
- Provided modeling guidelines, recipes and applied them to a real system model
- Reached the limits of SysML for systems engineering of
  - Requirements
  - Structure
  - Behavior

### Our current conclusion

- SysML can be used to model ground based astronomy domain
- 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 systems engineering stakeholders is the most important value