



Some experiences from Model Based Systems Engineering (MBSE)

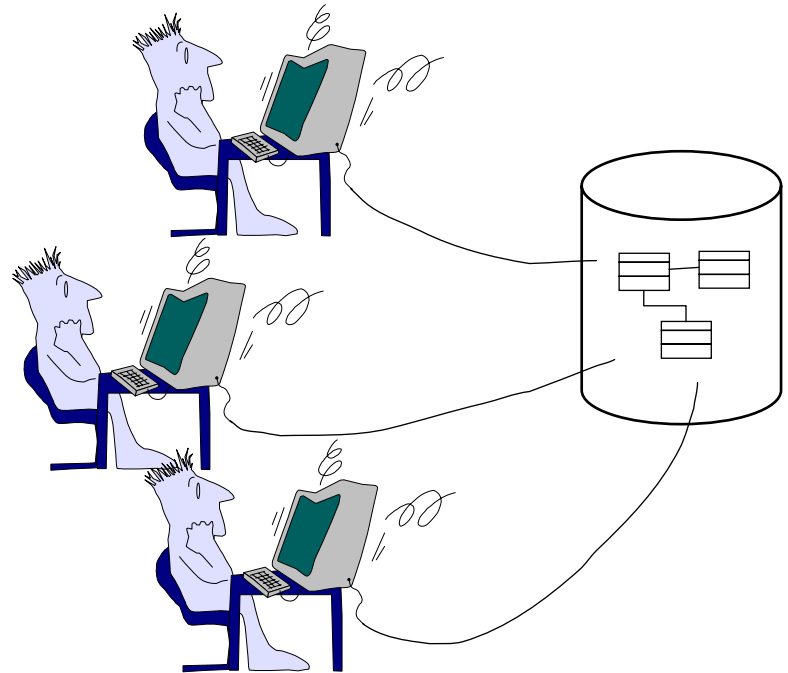
Anders Mattsson

Combitech Systems AB

anders.mattsson@combitechsystems.com

MBSE?

- You work on a common (mostly) graphical model of the system, where all work products are formally related to each other.
- The model is executable which allows testing on model level.
- Code and Documents (if used) are generated from the model.

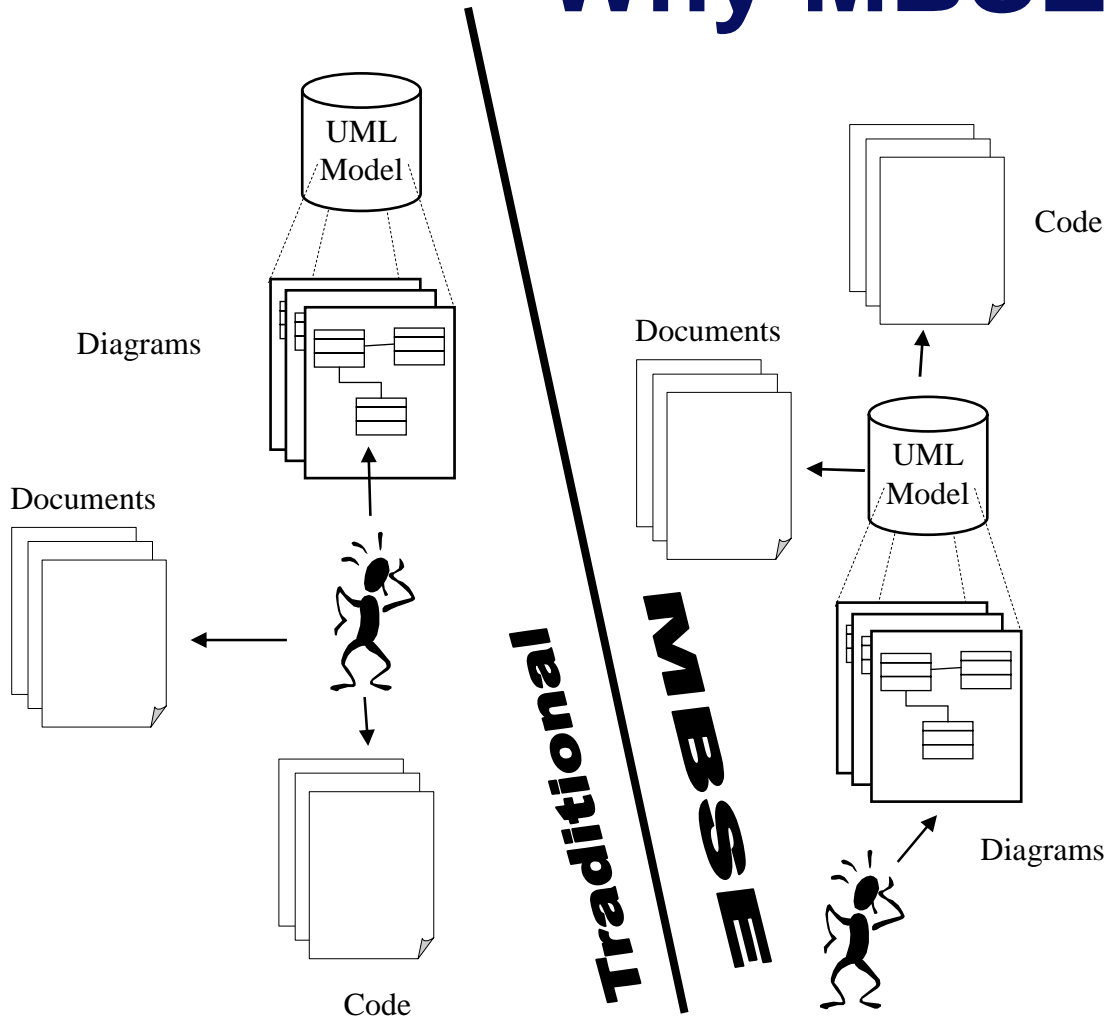




Implications

- No distinct separation of Analysis, Design, Implementation and Test.
- Tighter cooperation between developers.
- Configuration Management of one model instead of a number of documents.

Why MBSE?



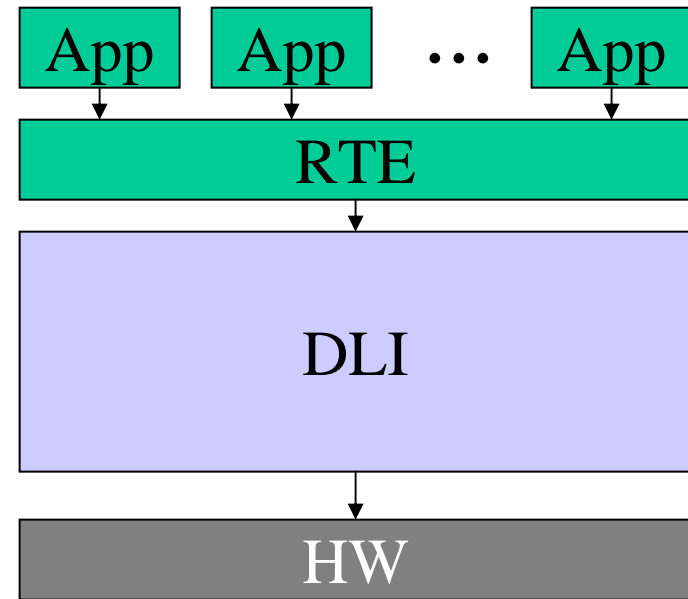
Less redundancy!

- Higher Productivity
 - Less work
- Fewer Errors
 - Fewer error sources
- Less Risk
 - Earlier, more frequent test
- More Creativity
 - Fewer non creative tasks



The System

- Digital TV set top boxes
- CS make the DLI, Device Layer Interface.
- On top of this is a virtual machine (RTE) that execute applications
- Special purpose HW, multiple processors
- DLI contains more than 2000 C++ classes, roughly 5MB machine code





The Project

- Roughly 40 SW developers during 18 Months
- HW is delivered very late in the project.
- Time To Market is crucial
- Large volume, low production cost.



Experienced benefits

- A feeling of working with the system, not with descriptions of the system.
- High confidence in development status.
- Testing is a natural part of construction on all levels.
- Less dependency on hardware.
- Very few errors at integration and system test
- High productivity, at least a factor 2.



Experienced Problems

- All parts in the model can't be correct and consistent with each other all the time!
- How does the tool handle large models?
- How do you navigate in a model consisting of thousands of classes and diagrams?
- How do you control a project where everyone is working on the same model?
- How do you get different tools to work together?

➤ **How do you avoid CHAOS ?**



You need more than a tool!

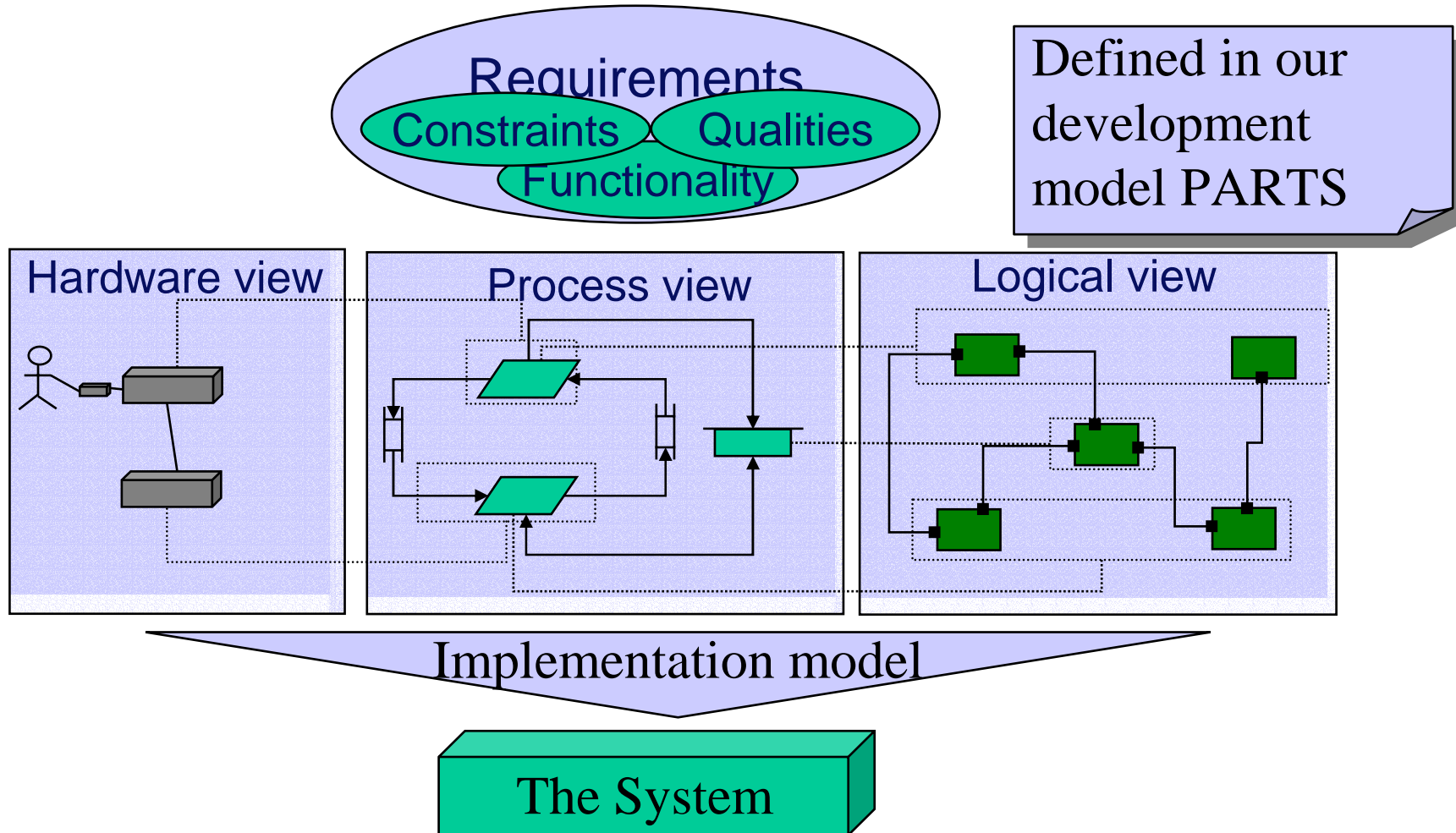
- People skilled in OO modeling
- A clear and appropriate architecture
- A detailed model structure (a model of the system models).
- A process defined in terms of that model structure.
- The tools has to be properly set up and integrated with each other and the process



MBSE compatible development model

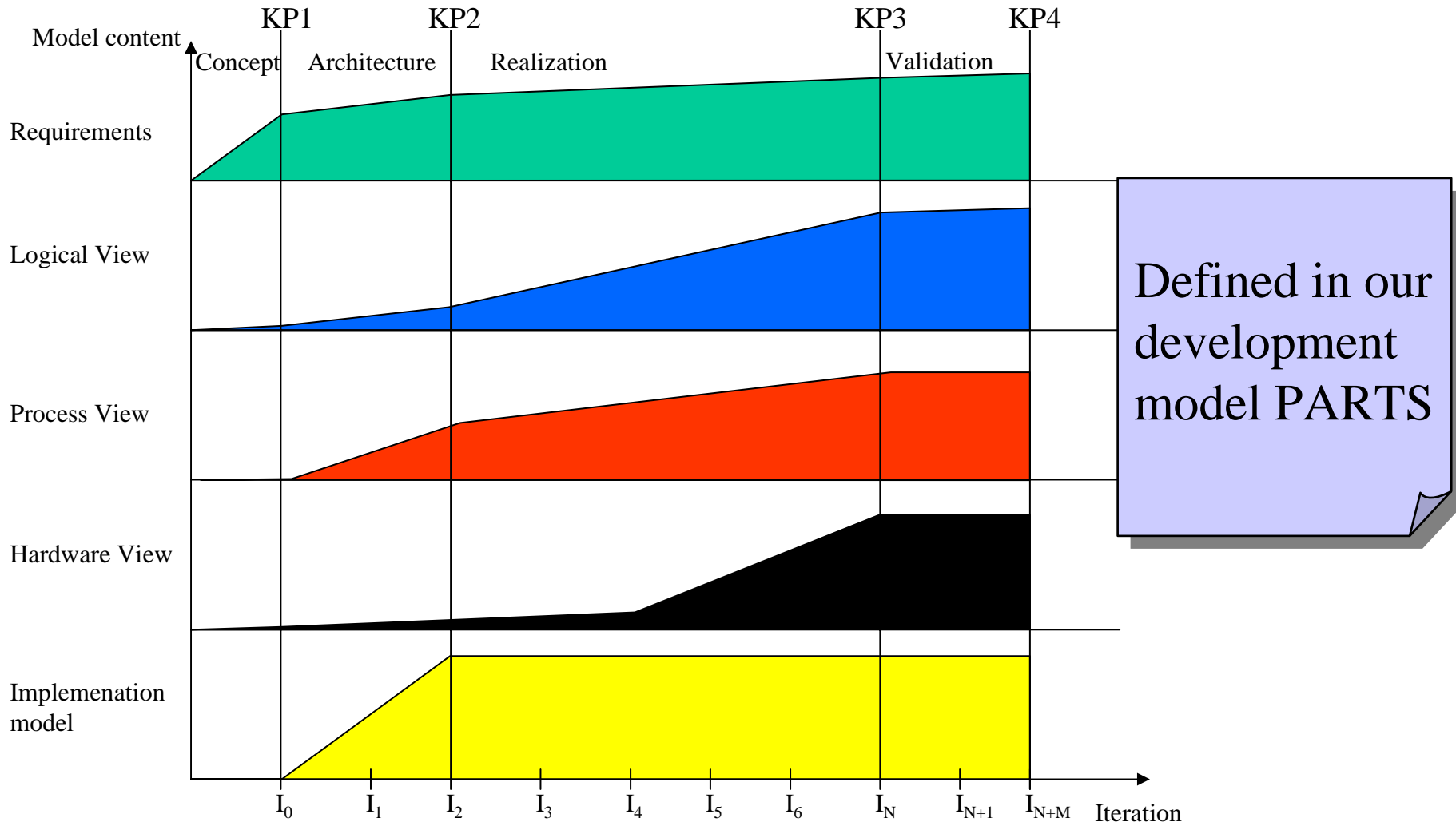
- Highly iterative, activities melt together, new activities.
- Milestones based on model status
 - Model structure must be defined
- Reviews on the model
 - Diagramming of the model must be defined
- New Roles

Our model structure





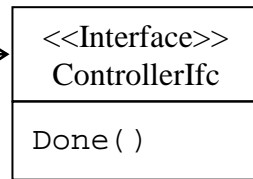
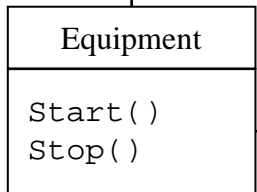
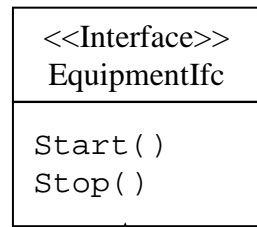
Our model based process



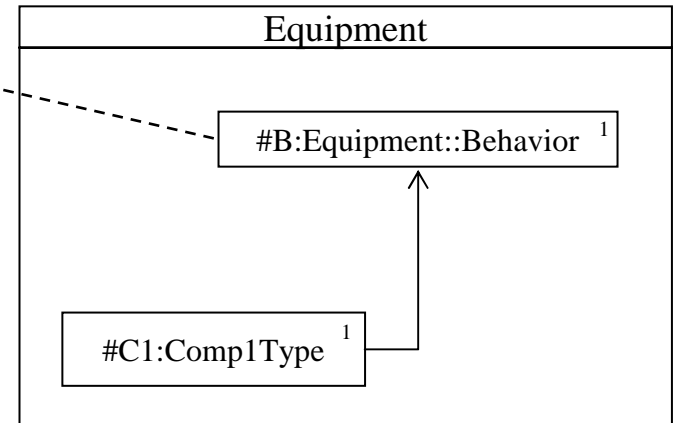


A Logical Component in UML

The Outside



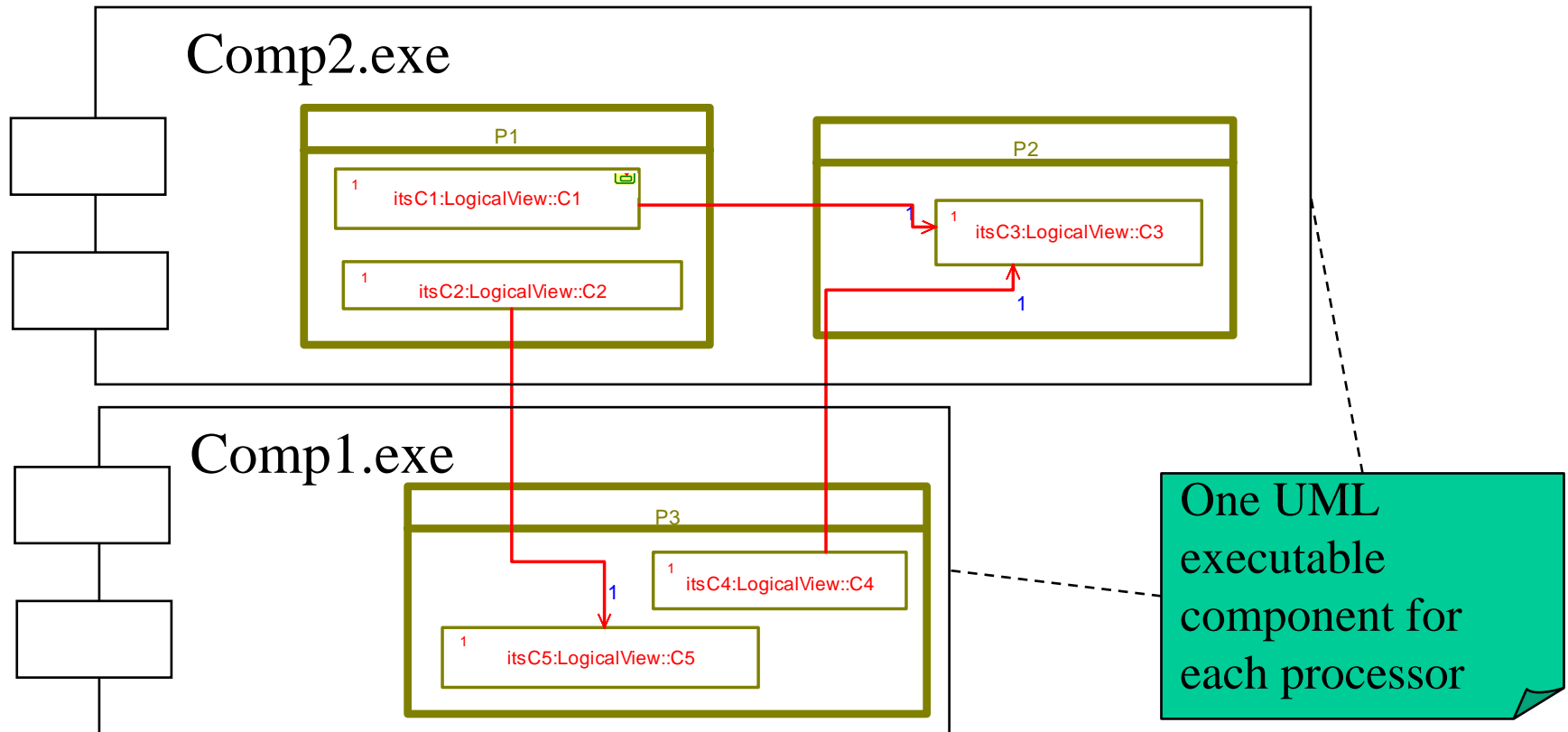
Hides the interface to internal components



The Inside

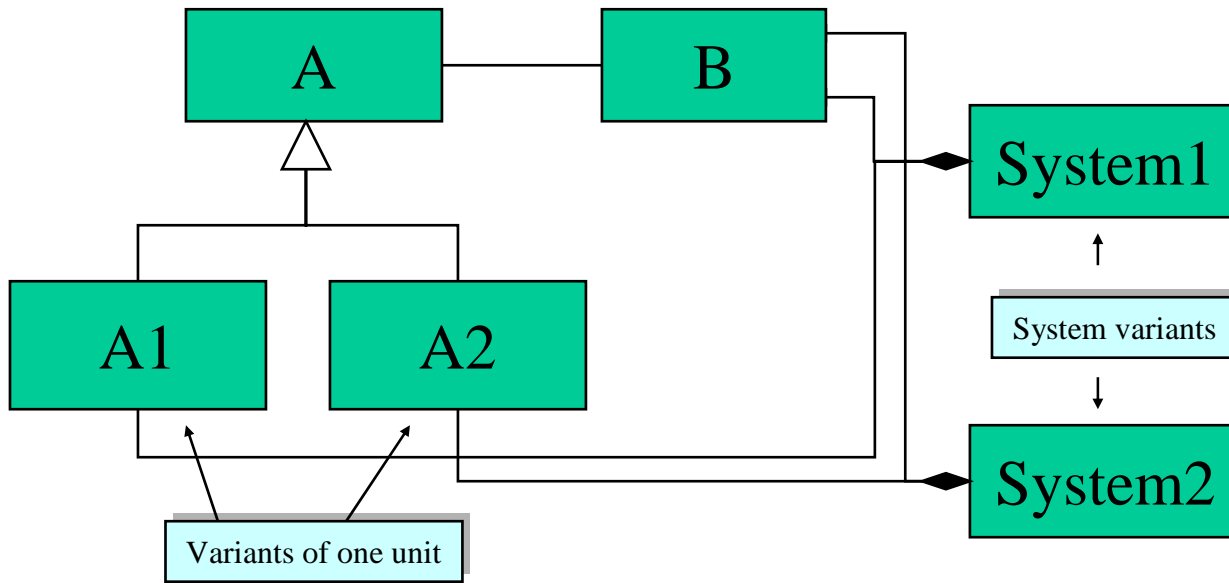


The Process View (in UML)

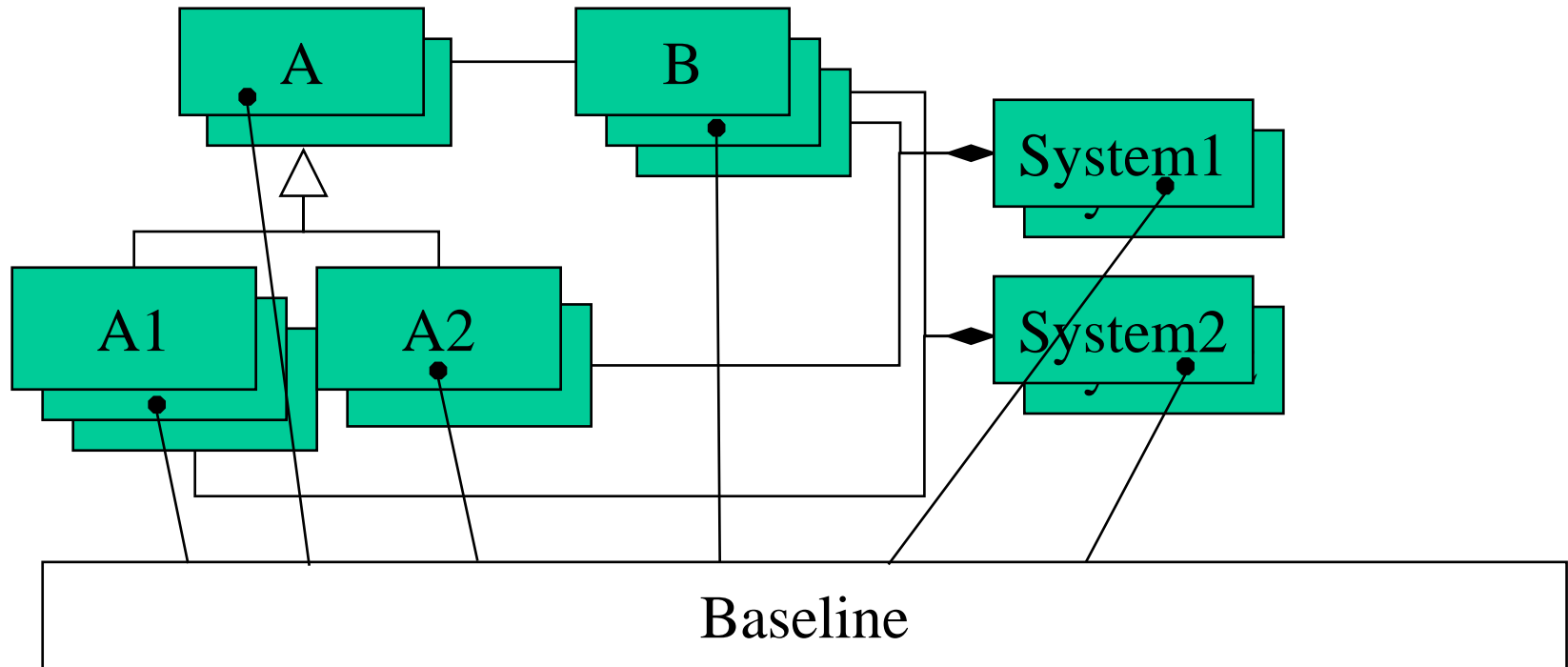




Modeling System Variants



- All variants in one model
- Variants are modeled as specializations
- The CM tool only handle versions, not variants
- A variant of the system is modeled as a composite class containing the right variants of different units



- The CM tool keeps track on all versions of units and systems.
- Baselines point out the versions of different units and system that makes up a certain version of the model.



Consequences

- Corrections to one variant is also made to all other
- You should coordinate tests on all variants.
- To save as much work as possible you must have as much common components as possible, this requires a thorough architectural work (product line architecture)



Contact me!

anders.mattsson@combitechsystems.com

0708-895 189