Specifying by Modelling?

Dr. Rudolf Hauber

HOOD GmbH
Büro München
Keltenring 7
D-82041 Oberhaching

Tel: 0049 89 4512 53 0
www.HOOD-Group.com

Presenter

- Dr. Rudolf Hauber
  - Senior Consultant
  - Responsible for Aerospace and Defense
  - Main emphasis: Modelling, SysML, UML
  - Rudolf.Hauber@HOOD-Group.com

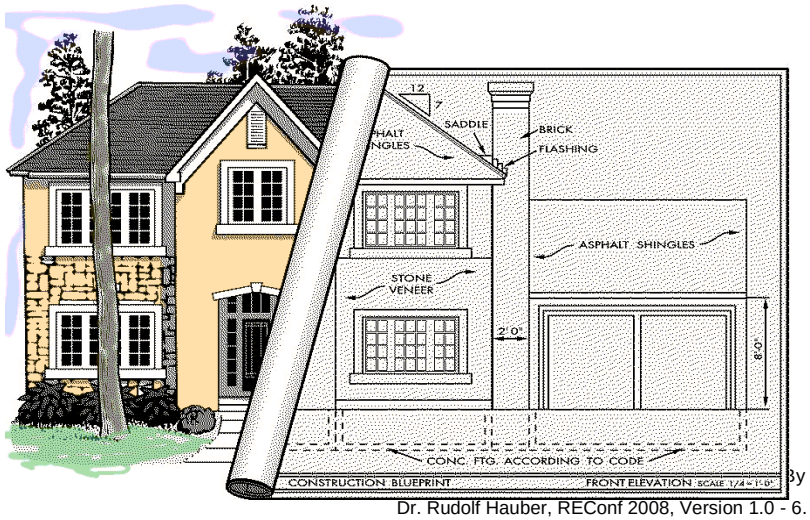
Content of this Presentation

Motivation

System and Software-development is not trivial!

- Representation of problem domain and solution using models
  - reduces complexity
  - simplifies communication
  - improves reuse

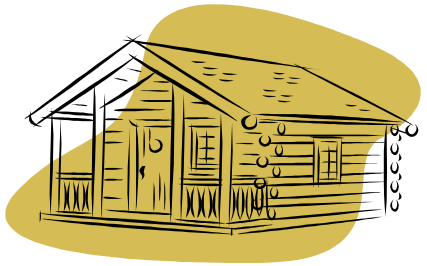
- Modelling is an established engineering technique

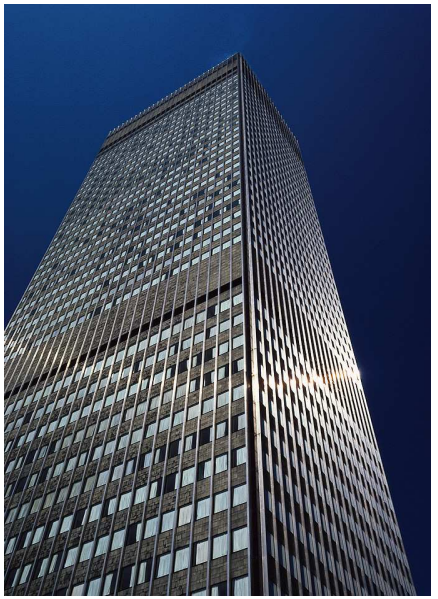


(Von Rational Websources)

Benefit of modelling

- Modelling is a tool for
  - Customer
    - Impression of product/system
    - Stabilisation of requirements
  - Project management
    - Risk mitigation by early verification (Requirements, Architecture, Design)
    - Cost reduction by automation
  - Development
    - Early proof-of-concept by simulation
    - Re-use of models
  - Quality assurance
    - Quality improvement by automation
    - Early testability
  - Other stakeholders
    - Communication tool
- Modelling is no end in itself!
  - Depending on project size and objectives





1 Motivation

2 System Development Process

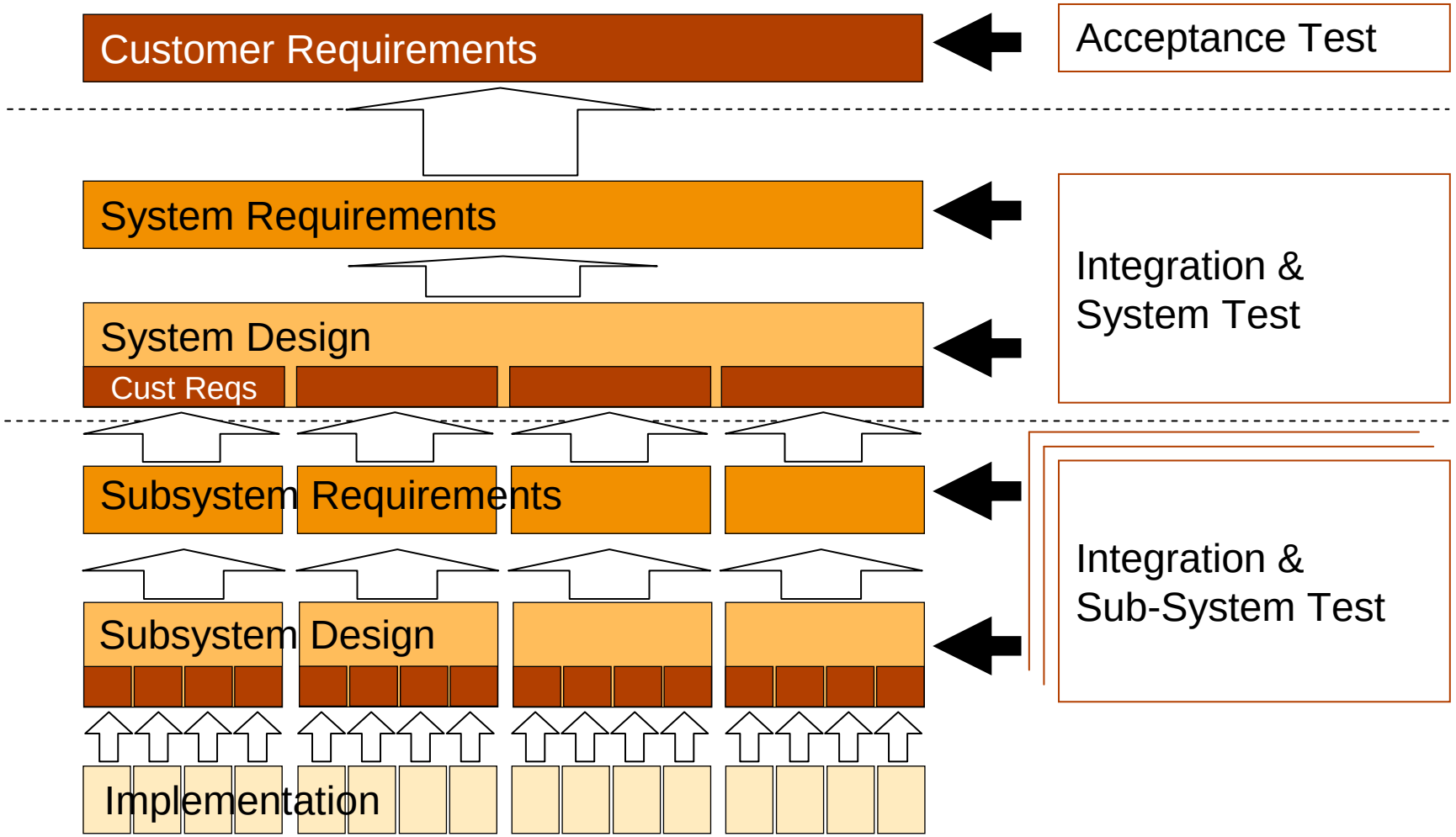
3 Modelling for Requirements Definition

4 RM and Modelling: Open issues

5 Conclusion

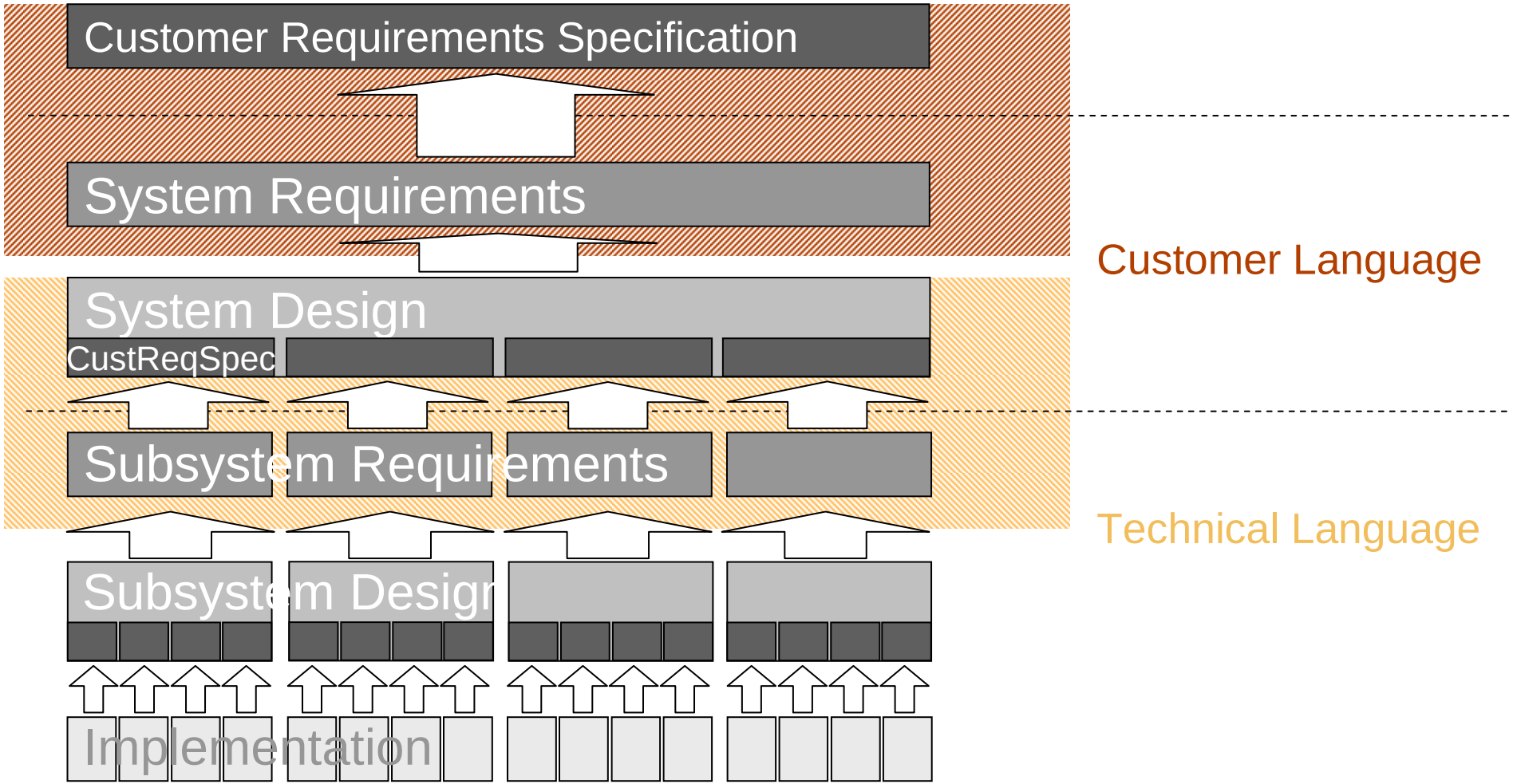
System Development Process

Systems consist of sub-systems and their interfaces.

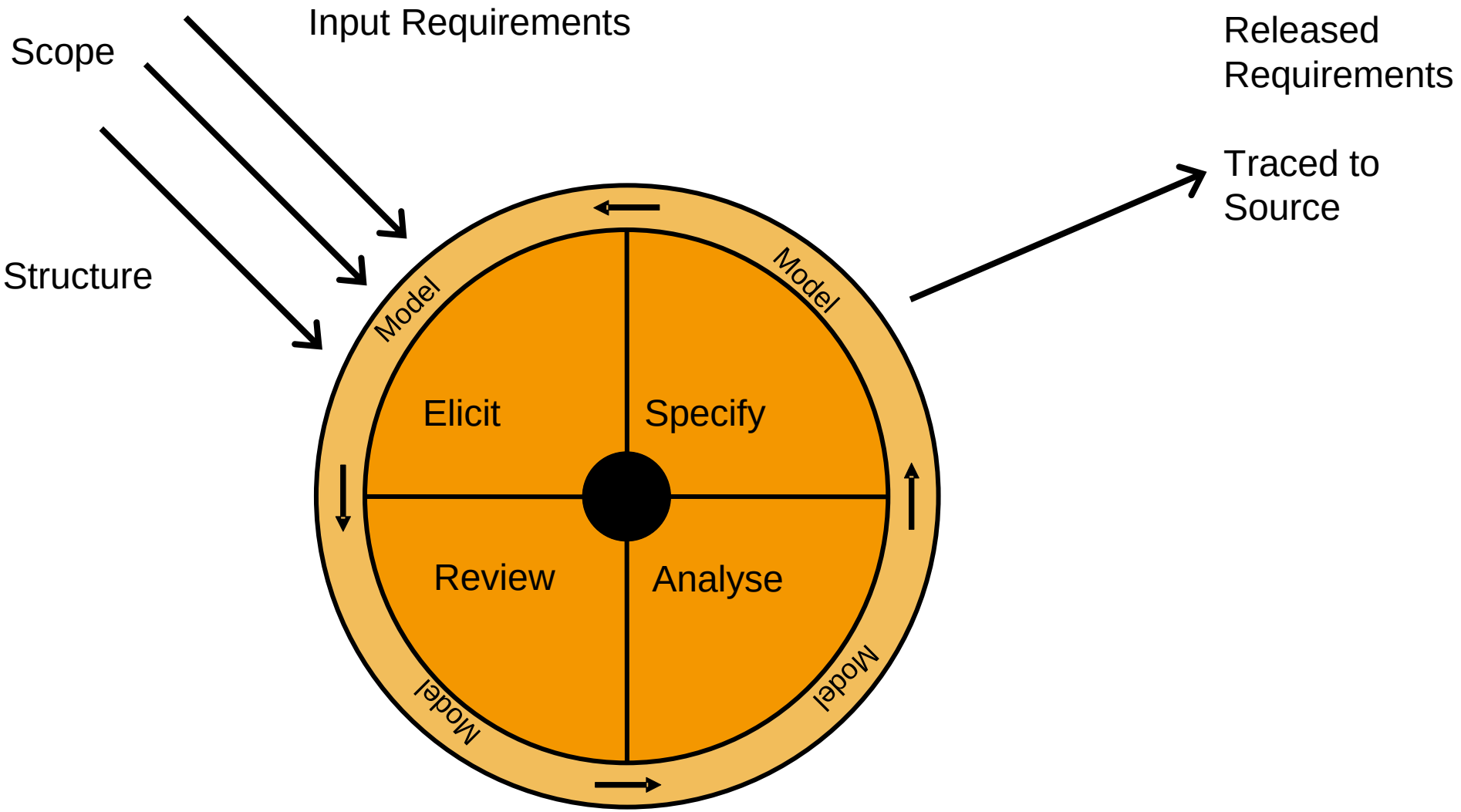


Customer Language and Technical Language

The difference between Design and System Requirements:

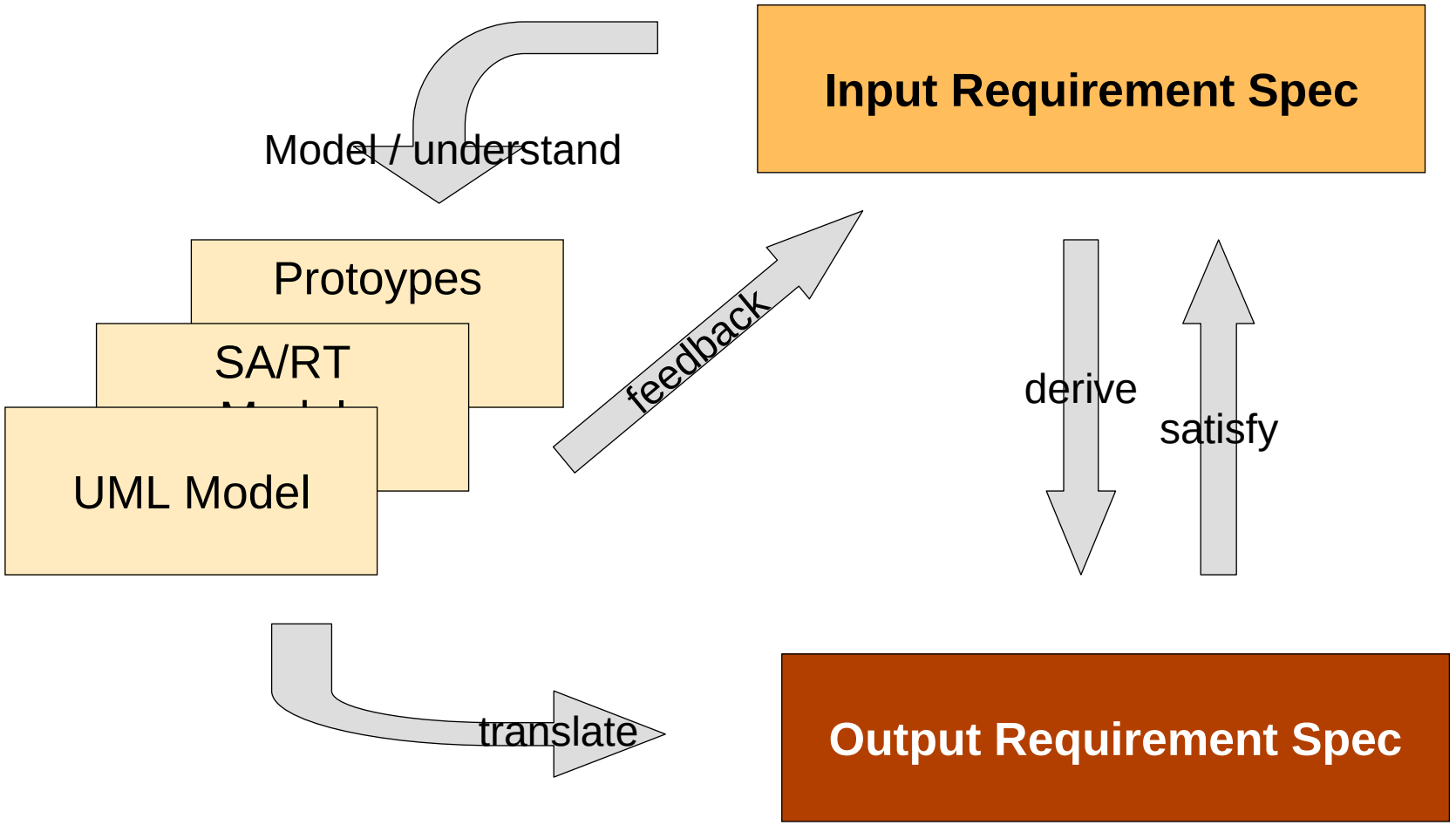


Requirements Definition Process

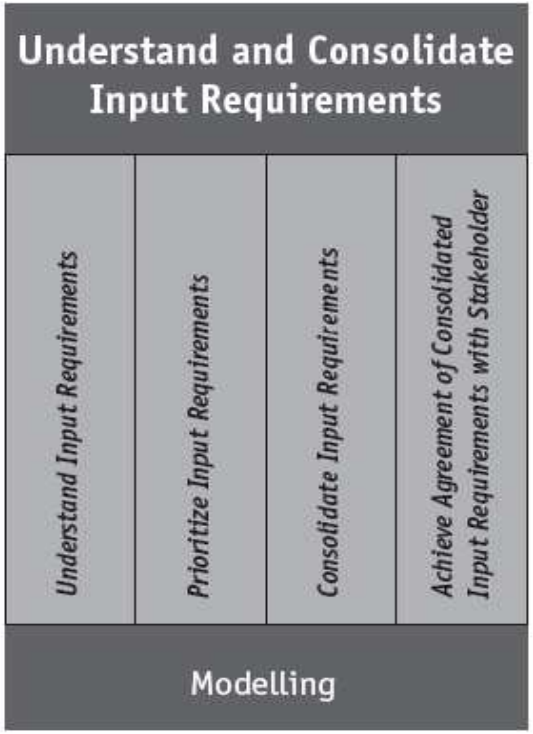


Requirements Definition – Modelling

Modelling supports all activities



Requirements Definition – Process activities

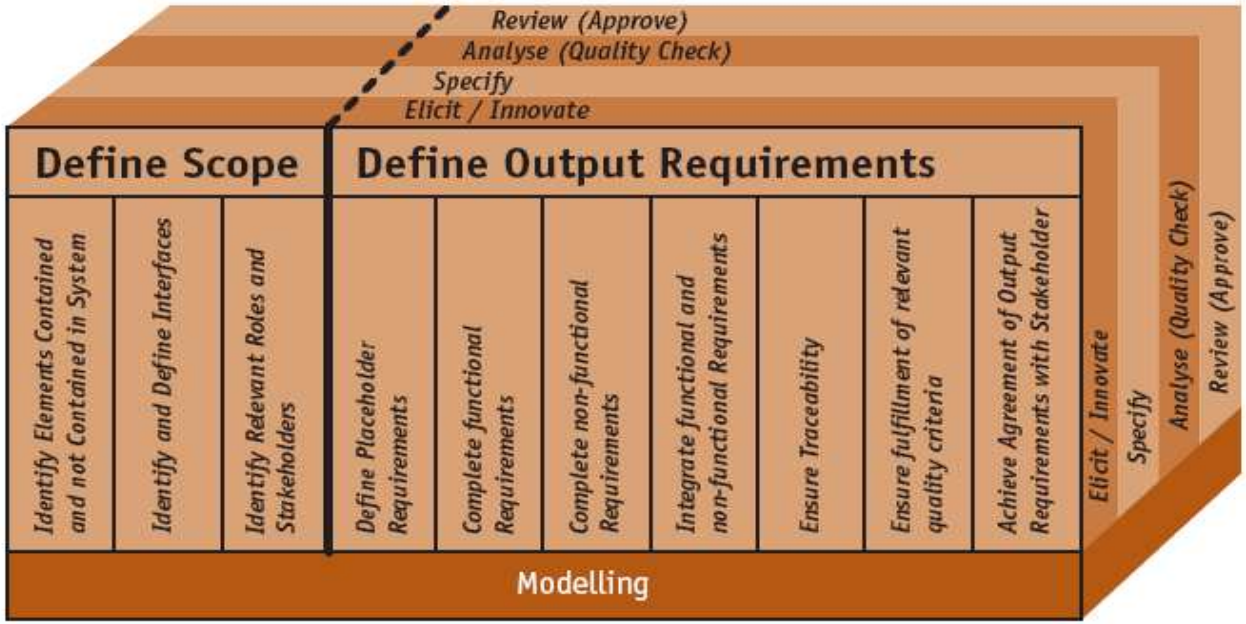


Understanding Requirements

Before we can define the next level of requirements we have to understand the requirements we seek to fulfill

Defining Requirements

The Requirements Definition Process is far more than just using a template

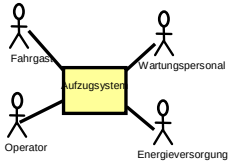


Source: HOOD poster „Erfolg ist Erfüllung der Anforderungen“, published 2007

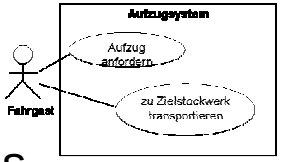
HOOD Modelling Process

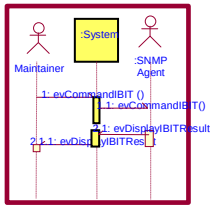
1. Understand top level requirements
   - Word, RM-database

2. Define System Scope, identify and specify interfaces
   - Context diagram

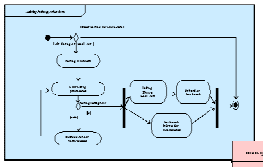


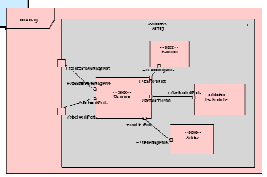
3. Define requirements
   - Identify and describe use cases using use case diagrams
   - Formalize use case flows using black-box sequence diagrams
   - Consolidate flows using activity diagrams



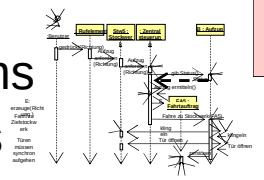


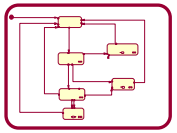
4. Define architecture
   - Find candidate architecure using internal block diagrams



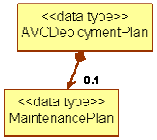


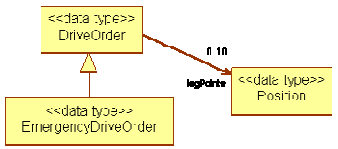
5. Specify subsystem requirements
   - Refine use case flows using white-box sequence diagrams
   - Consolidate subsystem responsibilities using state charts
   - Specify data flow using block definition diagrams





6. Consolidate subsystem requirements
   - Animate the model





1 Motivation

2 System Development Process

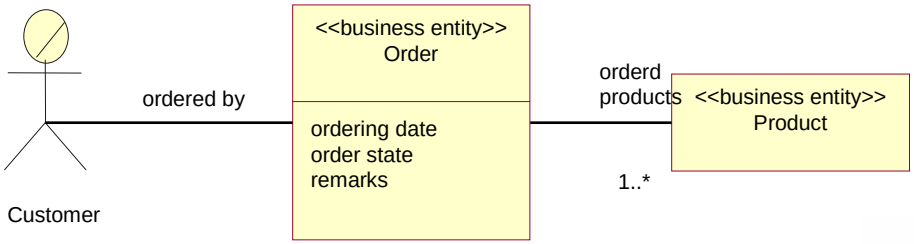
3 Modelling for Requirements Definition

4 RM and Modelling: Open issues

5 Conclusion

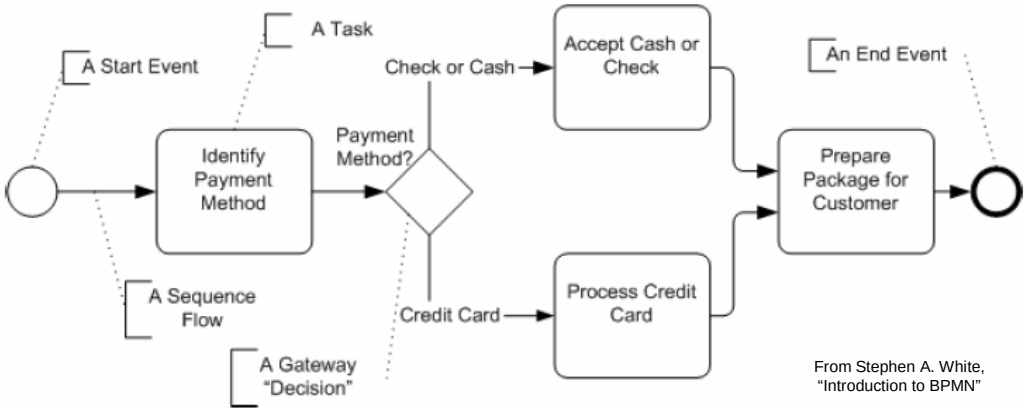
Customer Requirements

- Capturing current business situation
  - Modelling techniques can be very helpful
  - Business Modelling can help to understand current business:
    - Applying techniques like scenarios, use cases, interviews,...
    - Using notations for business processes, roles, data



UML Business Modelling profile

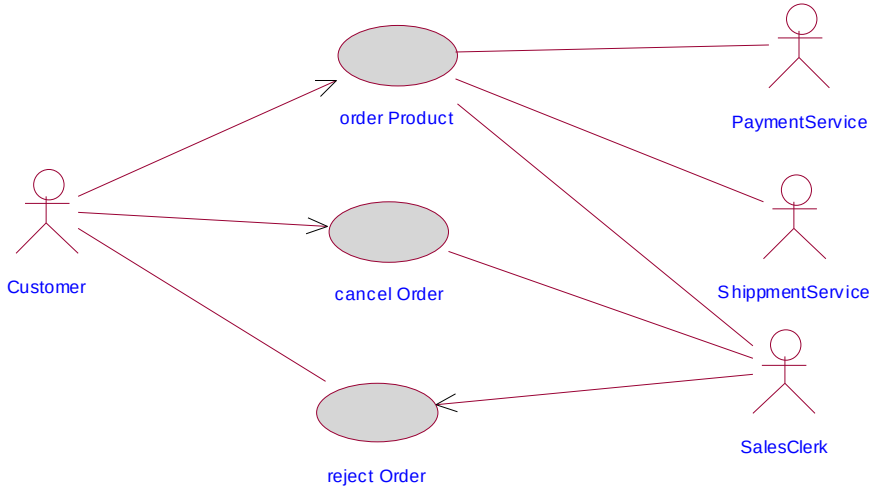
BPMN as new standard
(for BPEL as execution language)

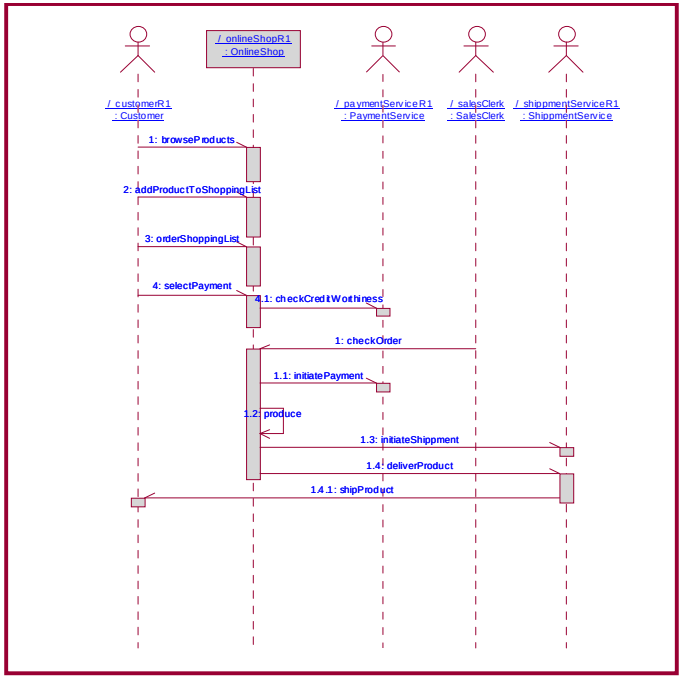


From Stephen A. White,
"Introduction to BPMN"

Customer Requirements

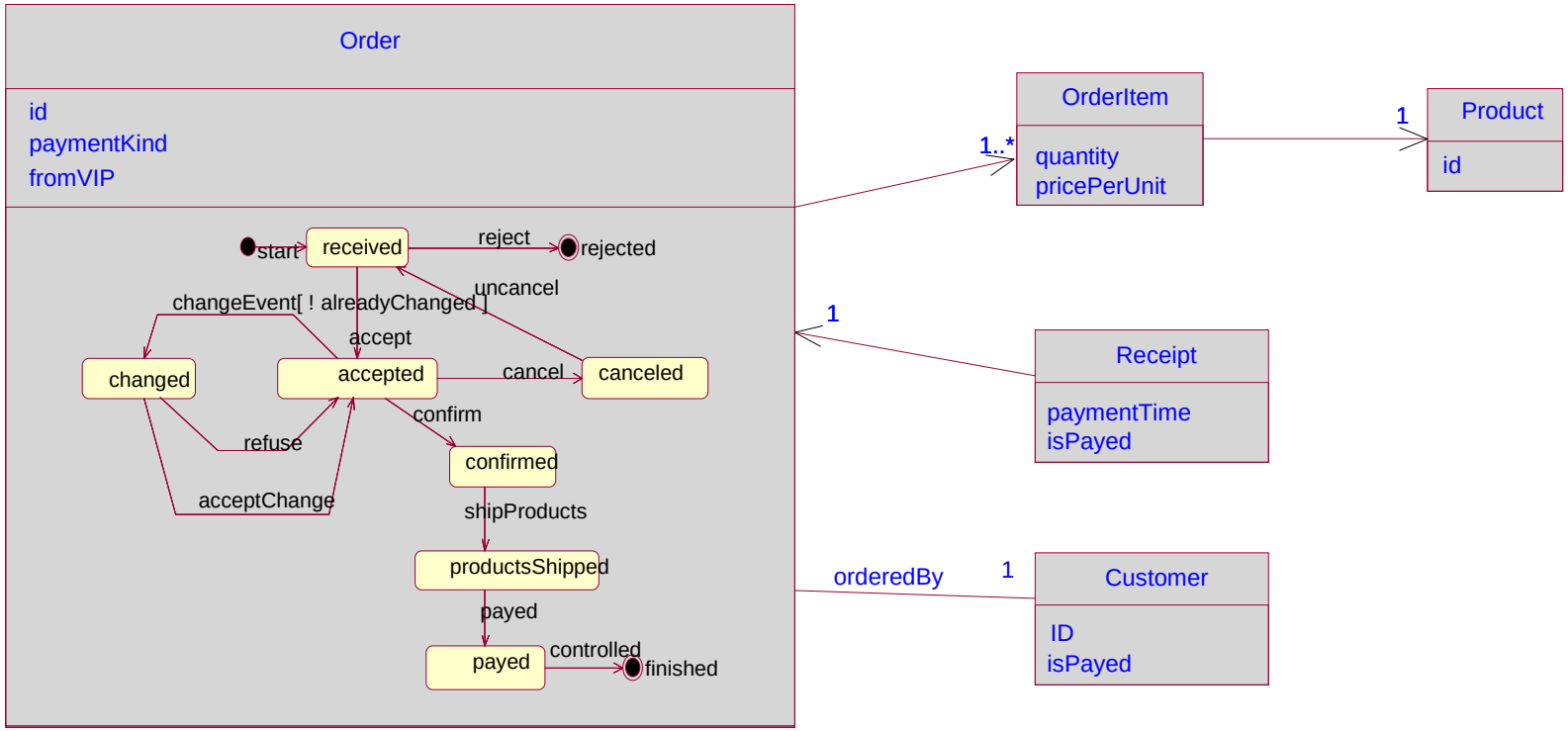
- Defining customer requirements
  - Use Case Modelling can be very helpful to understand a new system
  - Animation of user activities improves understanding of new processes
  - Using graphical notation to improve readability of requirements
    - „1 picture tells more than 1000 words“





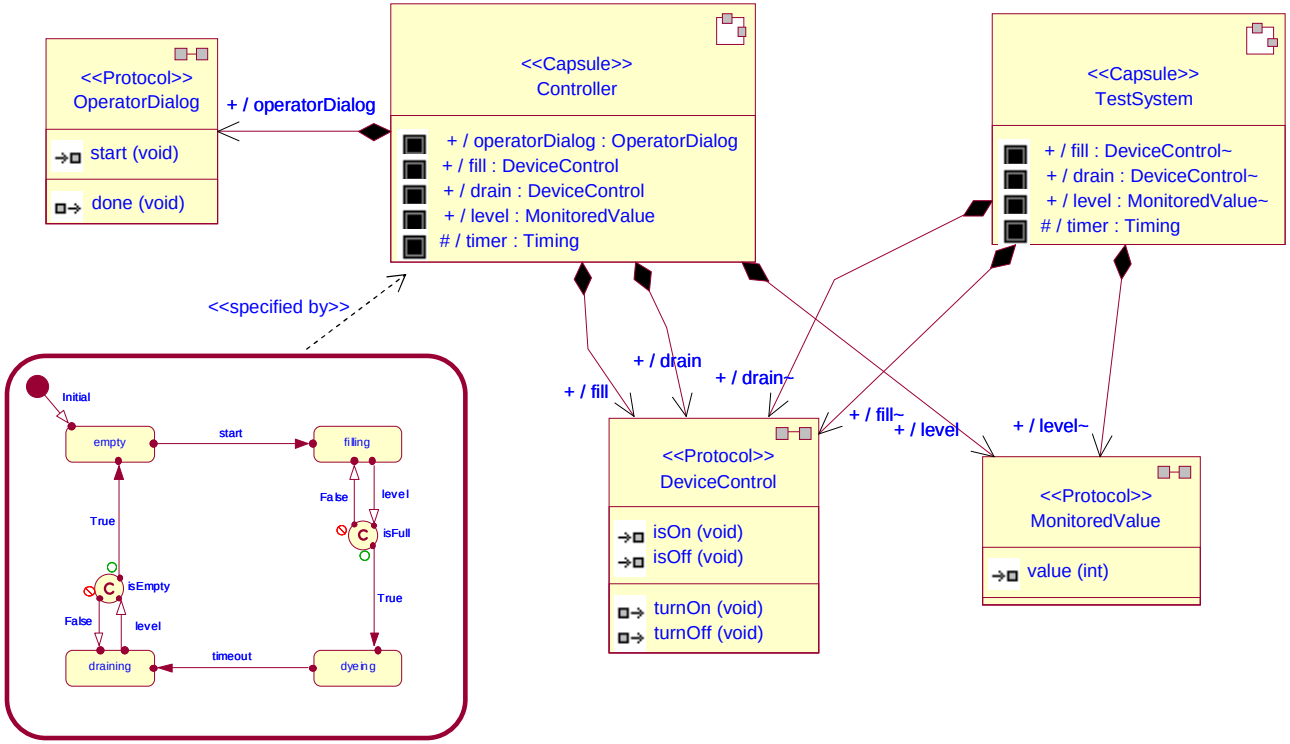
System Requirements

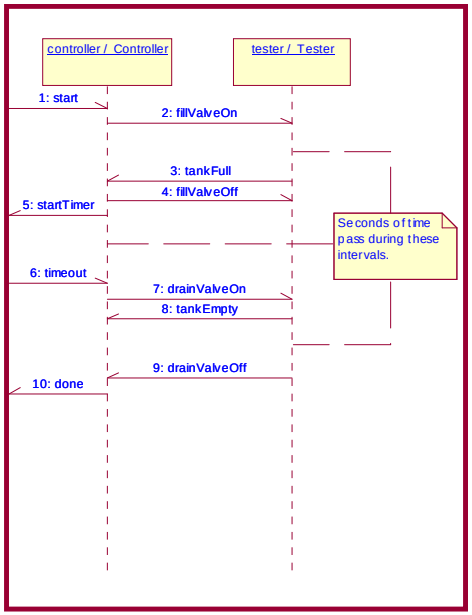
- Defining system requirements
  - Modelling can be very helpful to state customer requirements more precisely:
    - Formalised message flows using sequence diagrams
    - Business concept states & modes
    - Data Modelling
    - Timing



SW Requirements

- Defining software requirements
  - Modelling can be very helpful to derive SW from system requirements:
    - Precise interfaces/protocols
    - Subsystem modes and timing
    - Typed data models





1 Motivation

2 System Development Process

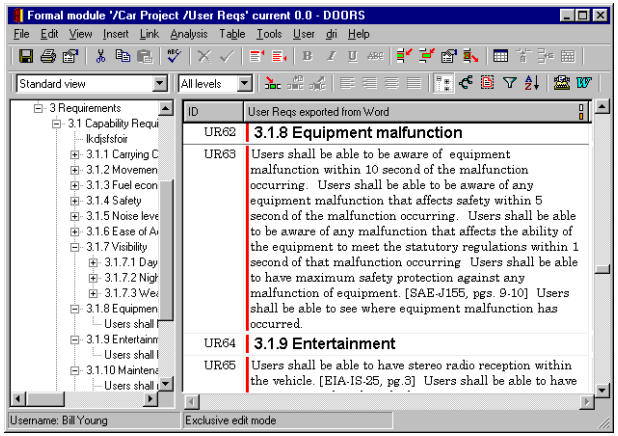
3 Modelling for Requirements Definition

4 RM and Modelling: Open issues

5 Conclusion

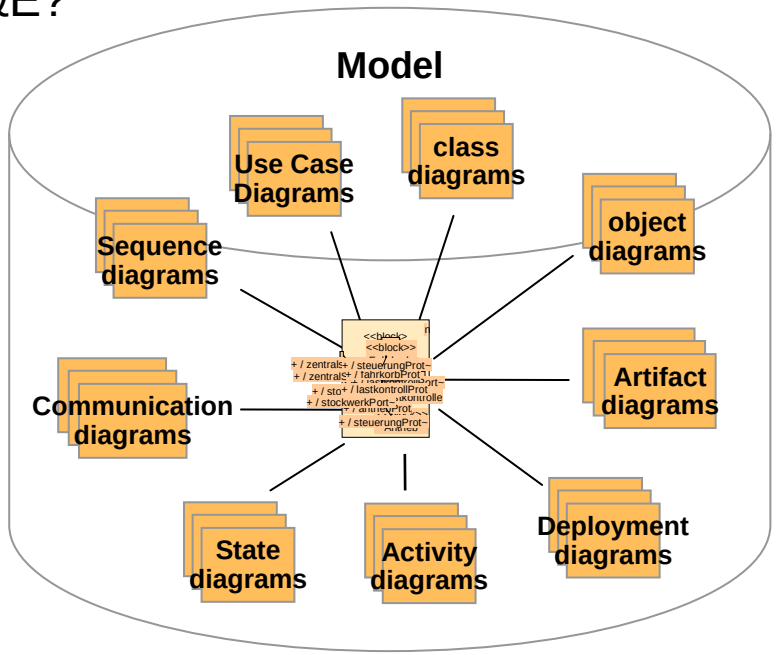
RM and Modelling

- Modelling can help specifiying requirements on every level
- Questions:
  - Can models enhance traditonal text-based RM&E?
  - Can models replace traditional text-based RM&E?



Enhance?

Replace?



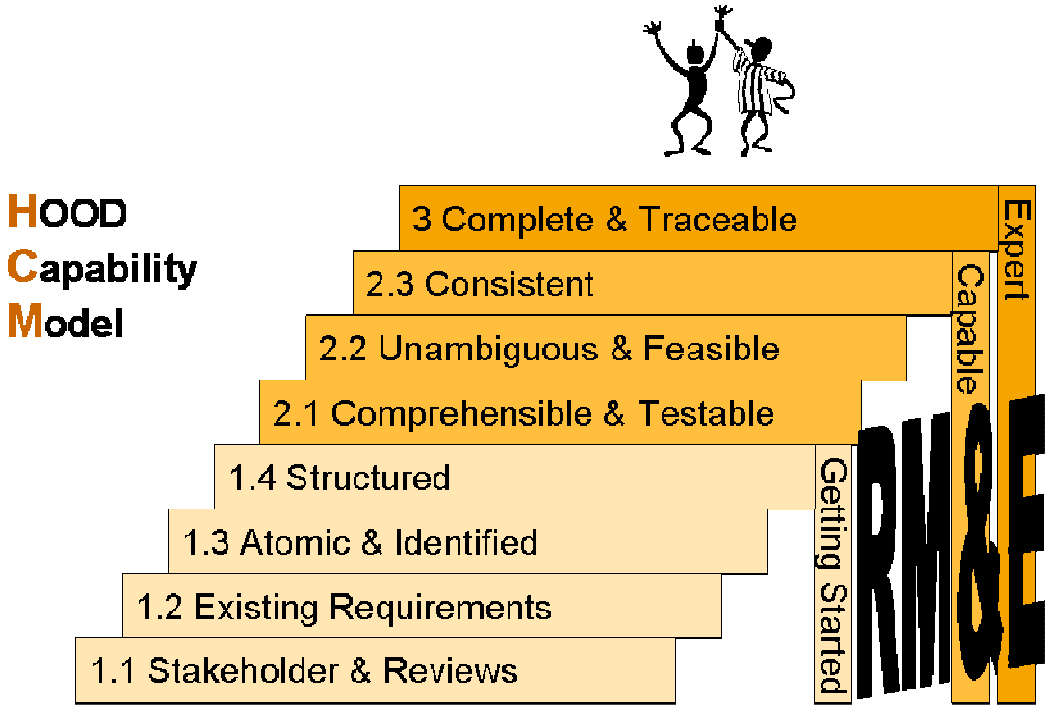
- New questions rise!

RM and Modelling: Open issues

- Is a diagram a requirement?
- Contains a diagram multiple requirements?
- Must a diagram be „translated“ into textual requirements?
- What‘s about traditional requirements quality criteria?

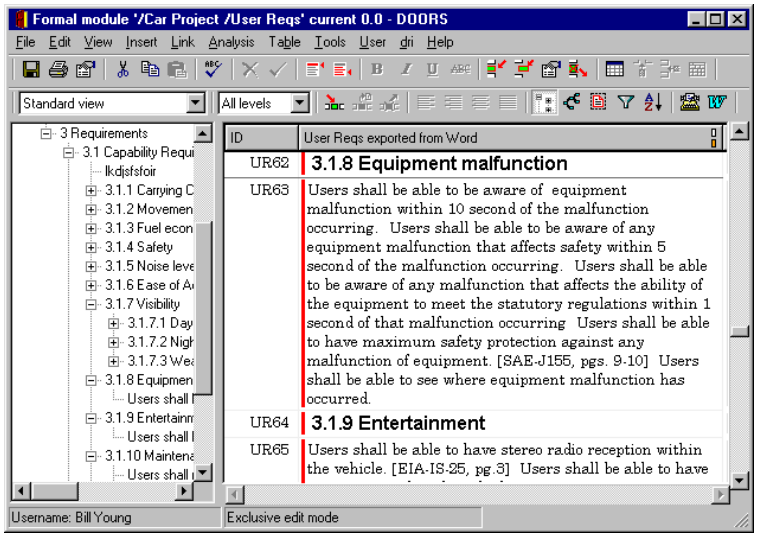
RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - identifiable
  - atomic
  - understandable
  - verifiable
  - non-ambiguous
  - feasable
  - traceable
  - consistent
  - complete
  - ...

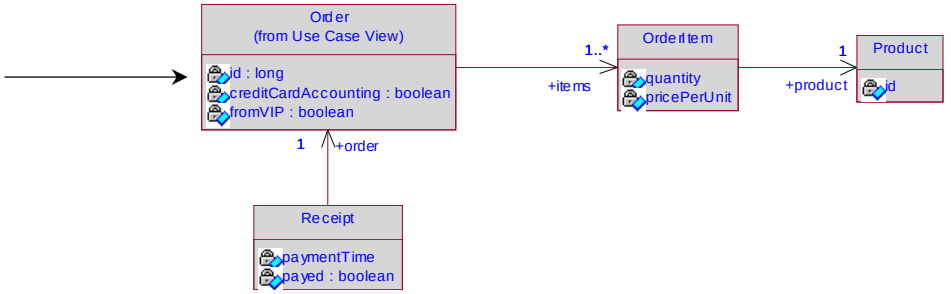


RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - **identifiable**
  - atomic
  - understandable
  - verifable
  - non-ambiguous
  - feasable
  - traceable
  - consistent
  - complete
  - ...

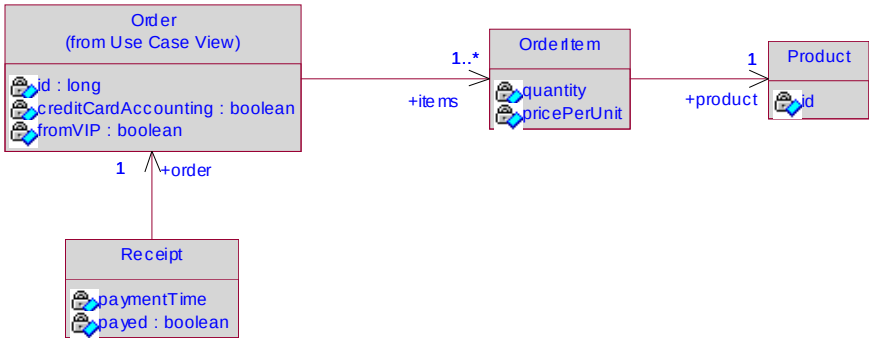


Internal model element ID

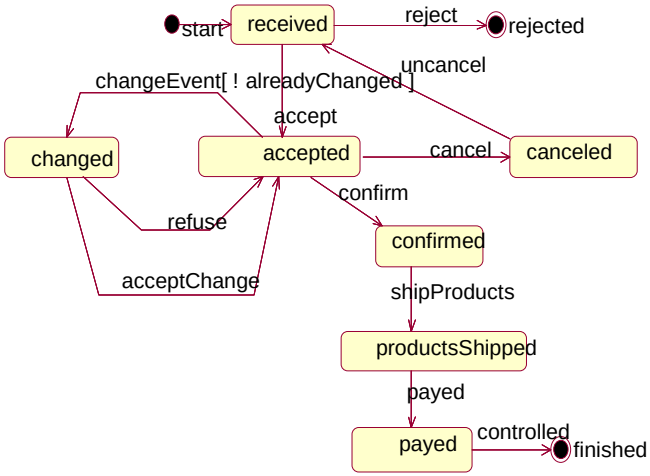


RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - identifiable
  - **atomic**
  - understandable
  - verifable
  - non-ambiguous
  - feasable
  - traceable
  - consistent
  - complete
  - ...



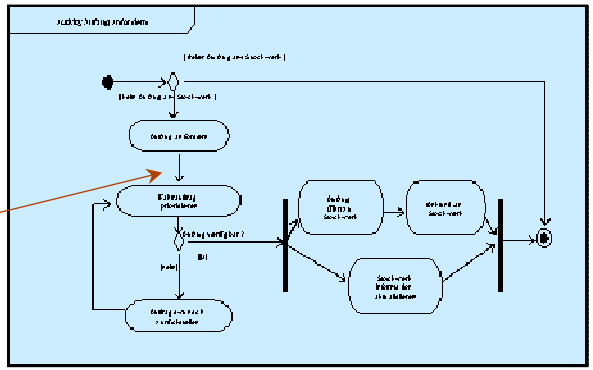
A diagram contains many requirements



Atomicity of Diagrams

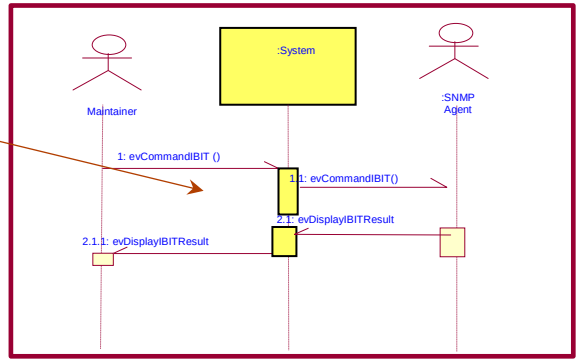
- State chart/activity diagram

Different flows



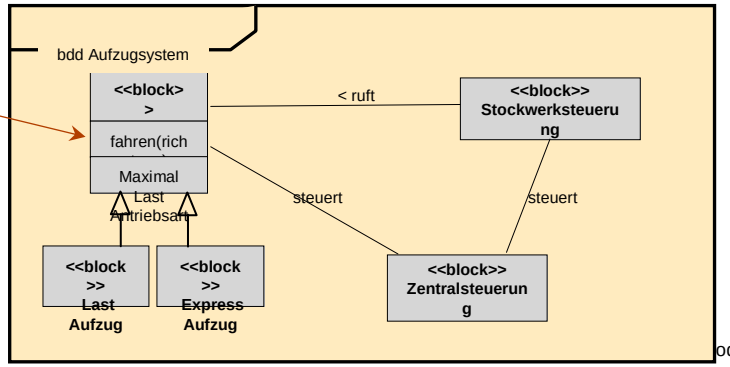
- Sequence diagram

Multiple service requests



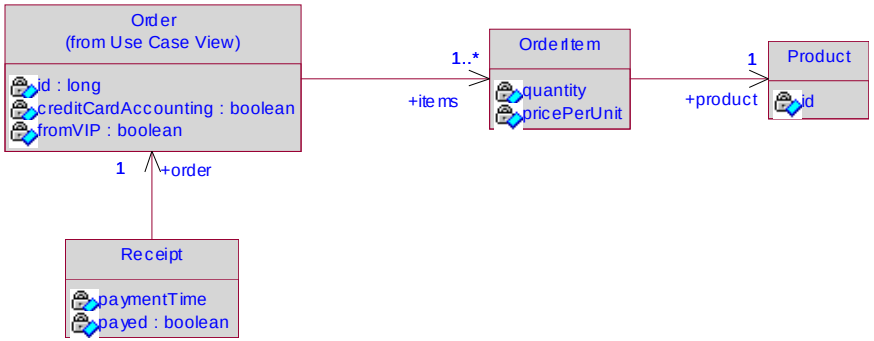
- Class/Block diagram

Multiple specifications

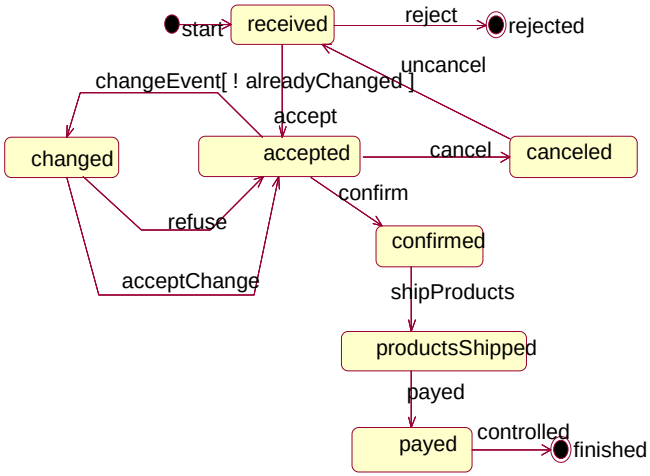


RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - identifiable
  - atomic
  - **understandable**
  - verifable
  - non-ambiguous
  - feasable
  - traceable
  - consistent
  - complete
  - ...

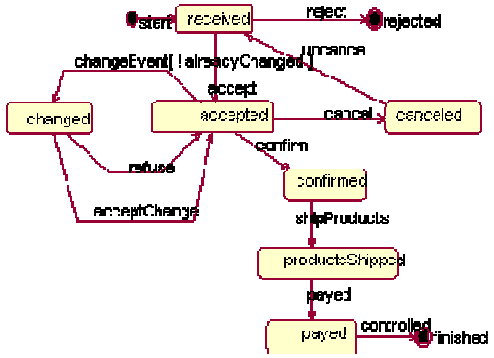


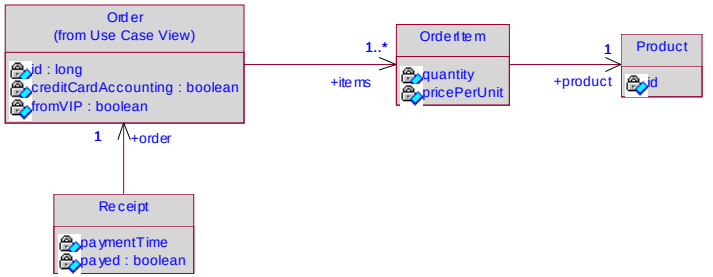
graphical notation improves readability

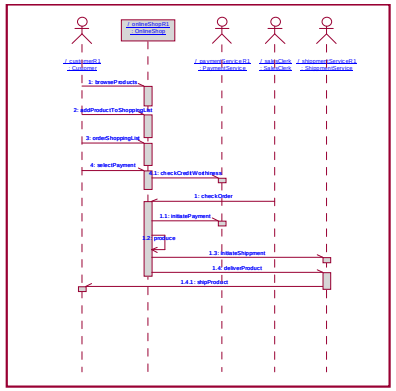


RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - identifiable
  - atomic
  - understandable
  - **verifable**
    - All test cases for state charts and activity diagrams can be derived automatically
      - see Binder Testing OO Systems
    - Class diagrams
      - CRUD test cases can be derived
    - For instance diagrams constraints are needed
  - non-ambiguous
  - feasable
  - traceable
  - consistent
  - complete
  - ...

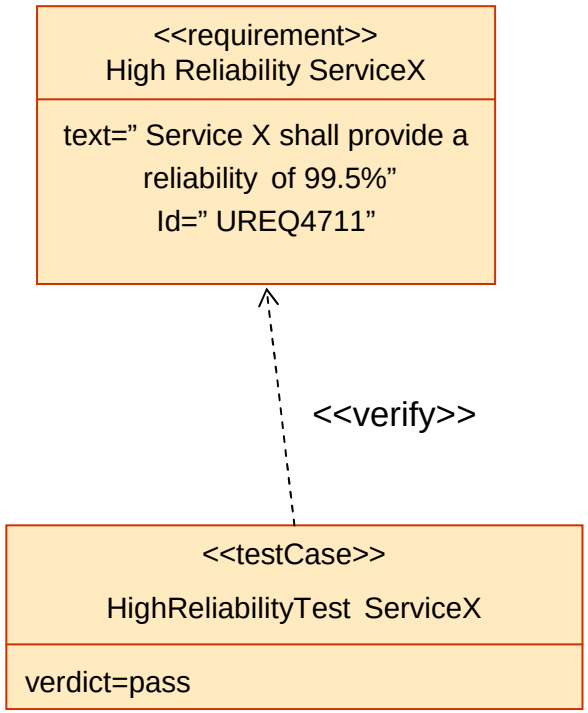


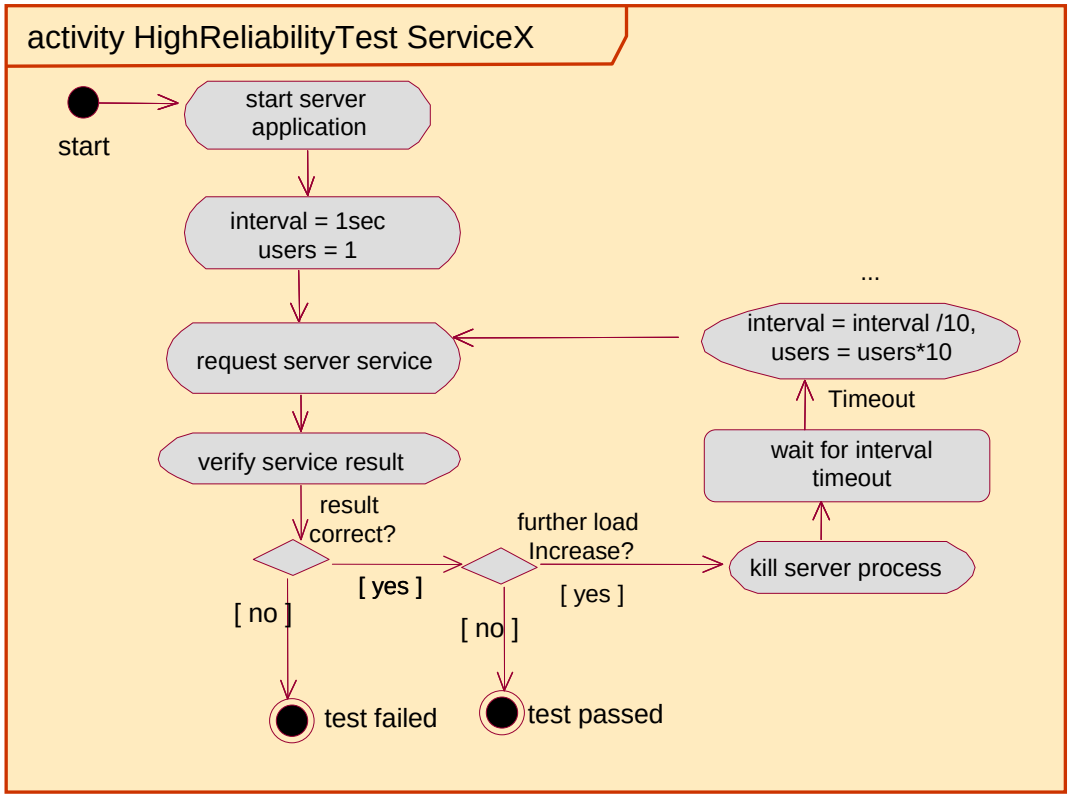




Verification

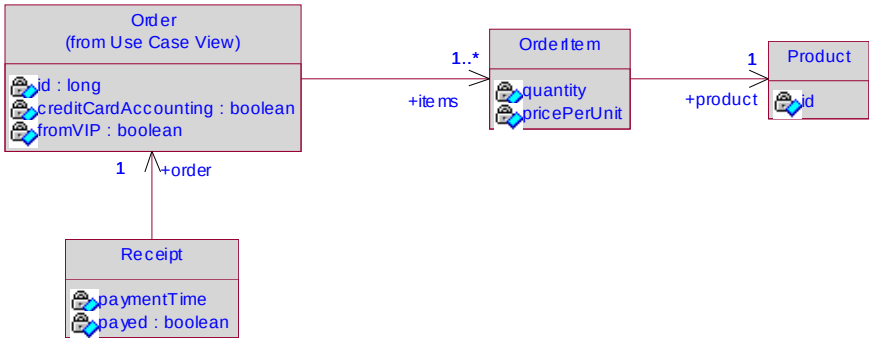
- Modelling can be used to define test models as well



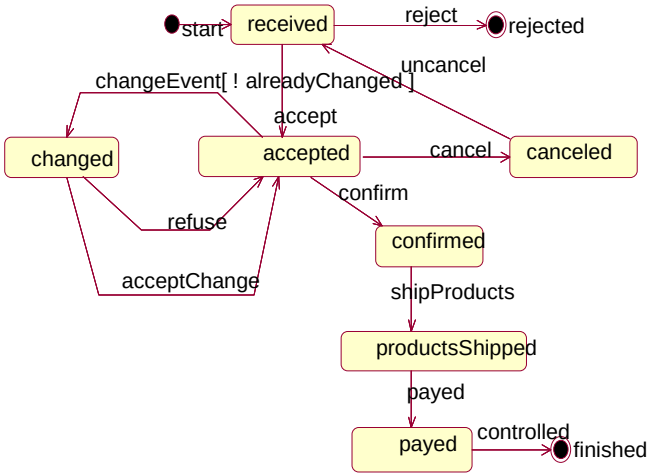


RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - identifiable
  - atomic
  - understandable
  - verifable
  - **non-ambiguous**
  - feasable
  - traceable
  - consistent
  - complete
  - ...

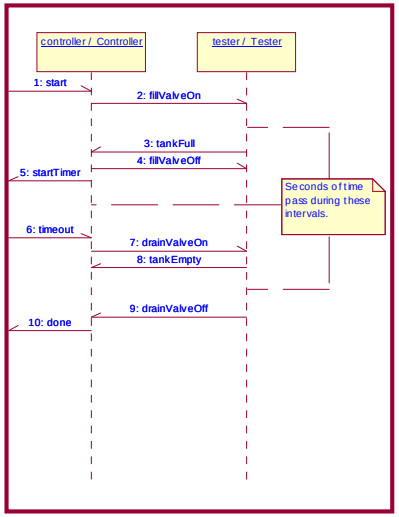


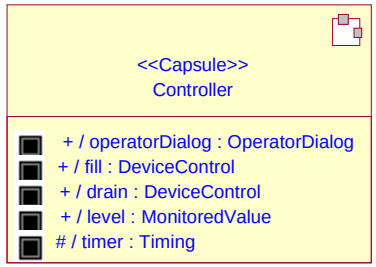
Precision is the „heart“ of formal languages



RM and Modelling: Modelling Limitations

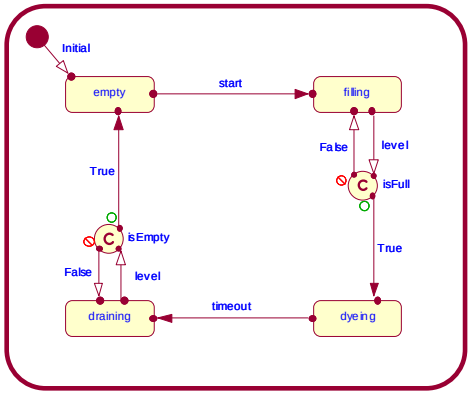
- Text-based requirements quality criteria:
  - identifiable
  - atomic
  - understandable
  - verifable
  - non-ambiguous
  - **feasable**
  - traceable
  - consistent
  - complete
  - ...





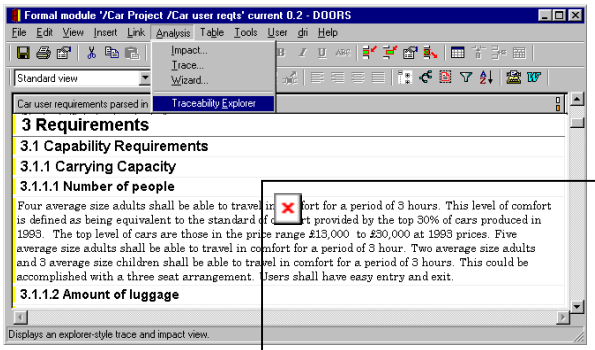
<<specified by>>

Model animation can be used as „proof-of-concept“

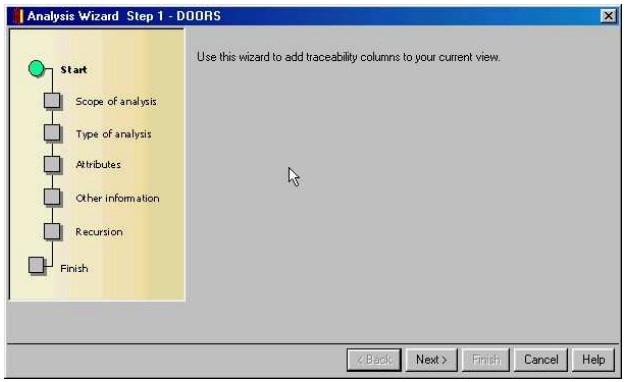


RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - identifiable
  - atomic
  - understandable
  - verifable
  - non-ambiguous
  - feasable
  - **traceable**
  - consistent
  - complete
  - ...

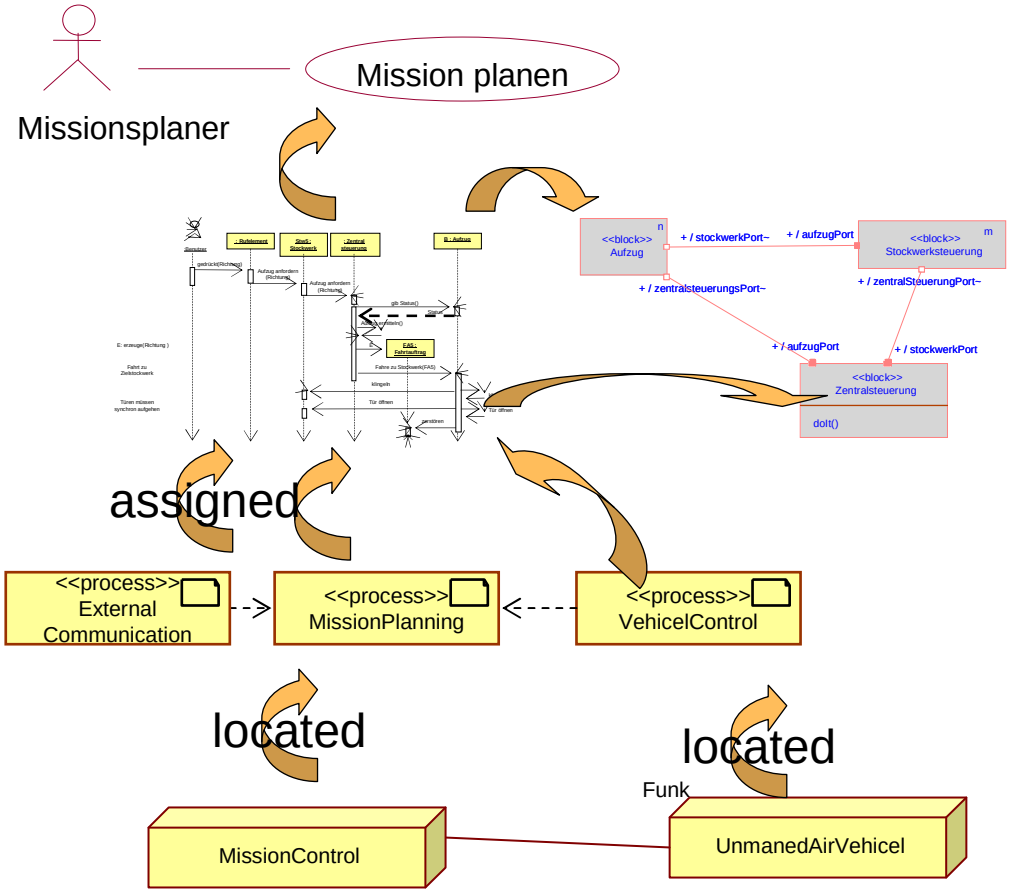


Tracebility features are the „heart“ of RM tools



UML/SysML traceability

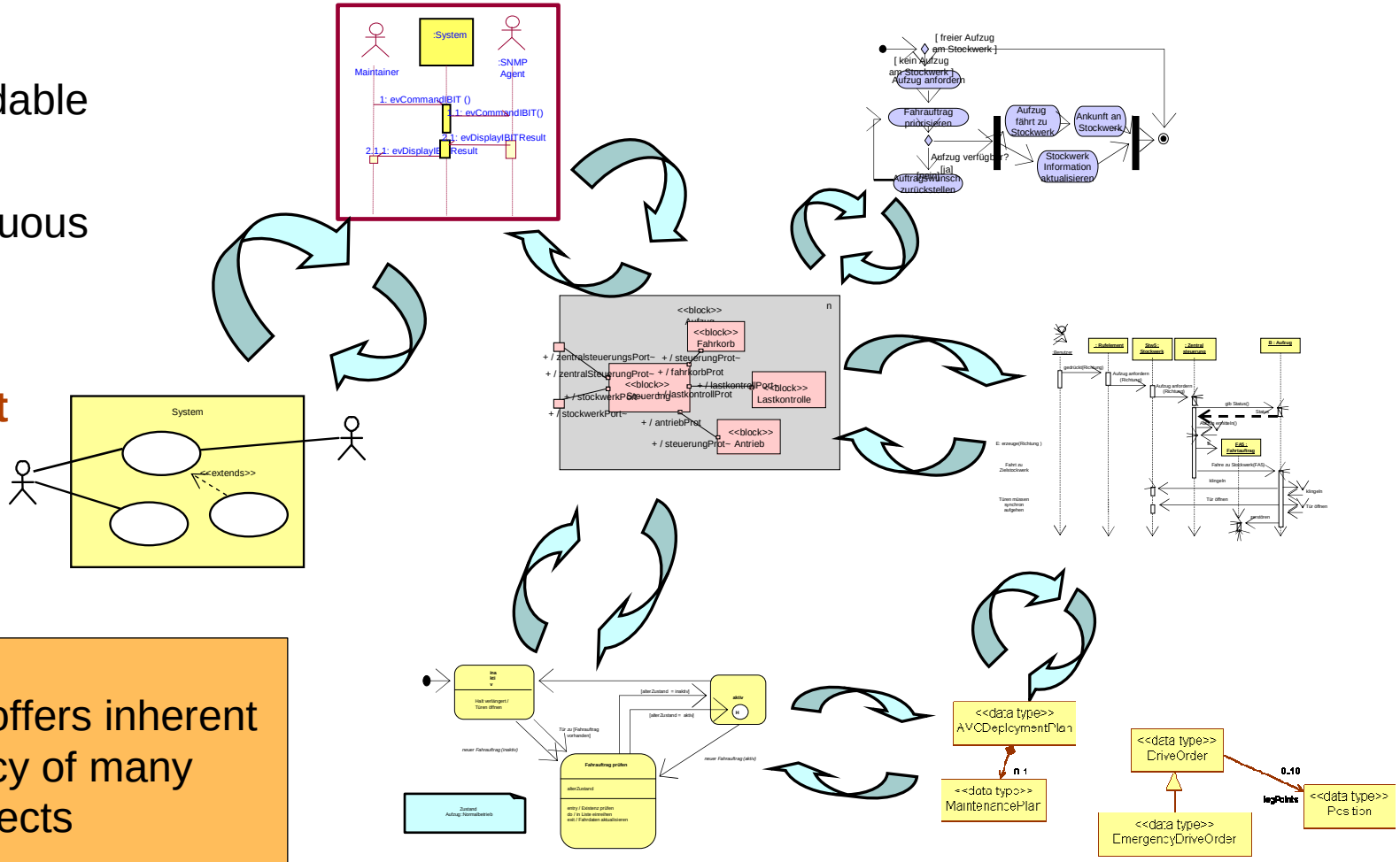
- **UML/SysML offers inherent traceability of many aspects**



Traceability analysis can be done using tool script

RM and Modelling: Modelling Limitations

- Text-based requirements quality criteria:
  - identifiable
  - atomic
  - understandable
  - verifable
  - non-ambiguous
  - feasable
  - traceable
  - **consistent**
  - complete
  - ...



UML/SysML offers inherent consistency of many aspects

RM and Modelling: Modelling Limitations

- Modelling is all about functionality
  - Flows
  - Timing
  - ....
- Non-functional aspects (Quality of Service) is hard to incorporate
  - Usability
  - Safety
  - Security
  - Reliability
  - Efficiency
  - Interoperability
  - Maintainability
  - Flexibility
  - Portability
  - Expandability
- Additional textual specification is still neccessary

1 Motivation

2 System Development Process

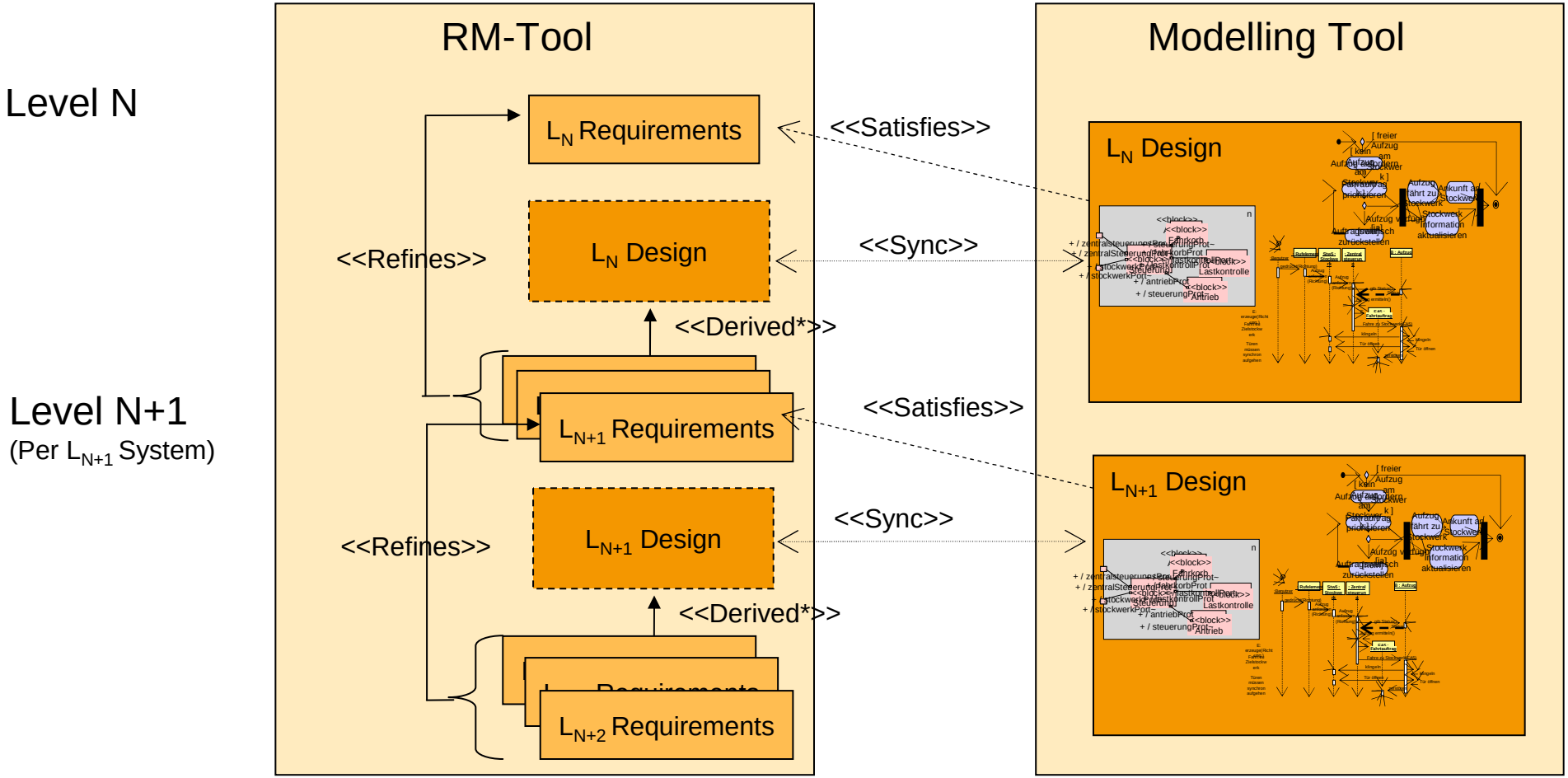
3 Modelling for Requirements Definition

4 RM and Modelling: Open issues

5 Conclusion

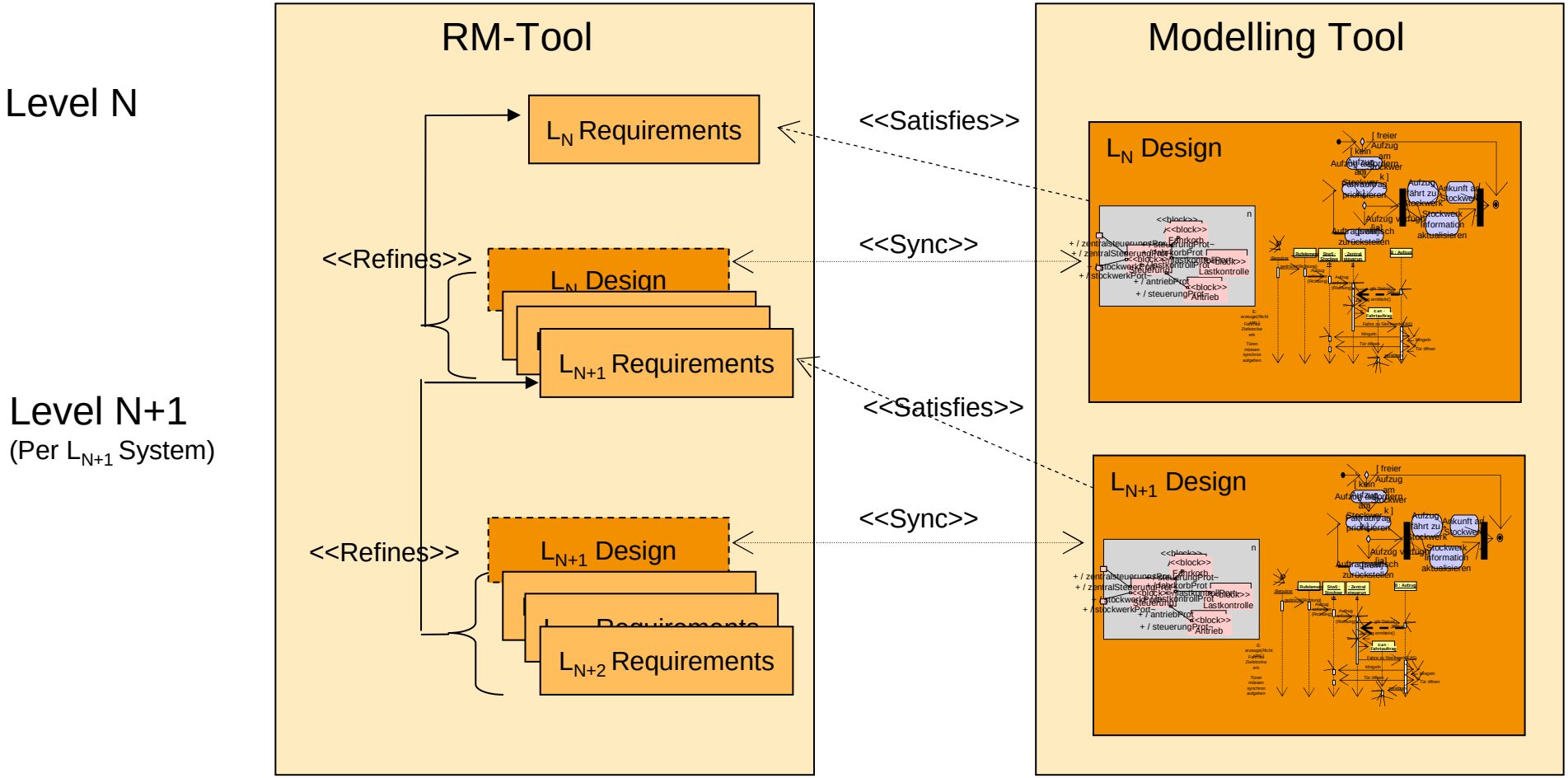
RM and Modelling: Summary

- Combine RM and Modelling:
  - 1. Scenario: Modelling to derive requirements



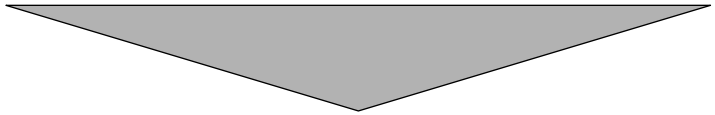
RM and Modelling: Summary

- Combine RM and Modelling:
  - 2. Scenario: Modelling enhances RM



Summary

- **Modelling is useful for specifying functionality!**
- **Consider customer expectations!**
- **Engineers must be trained in modelling**



There are many options

Just use modelling to gather requirements (Specification is still text-based)

Add diagrams to your Specification where useful

Develop a full requirements model