Modeling requirements

Slides used in the video available here
https://polimi365-my.sharepoint.com/:v:/g/personal/10143828_polimi_it/EfuNIeNgPwtIrwcJQNI1s2YBw6G0vEhQDw4yNbpqLH4wQg?e=per3xm

Modeling requirements

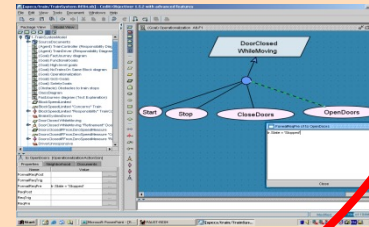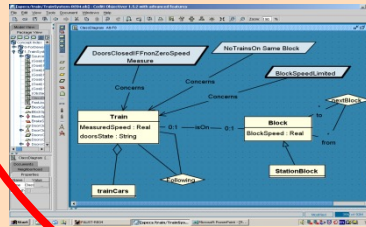- What is a model
- What to model in RE
- Tools for modeling

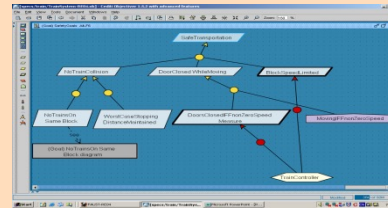# The Central Role of Requirements Models

stakeholders

existing systems

documents

elicitation & **modelling**

Requirements Models

analysis & validation

generation of RE deliverables
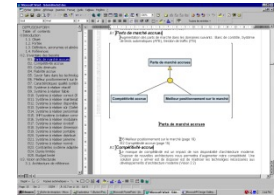
requirements document

Model: A definition

*"A model is a representation in a certain medium of something in the same or another medium.
The model captures the important aspects of the thing being modeled and simplifies or omits the rest"*
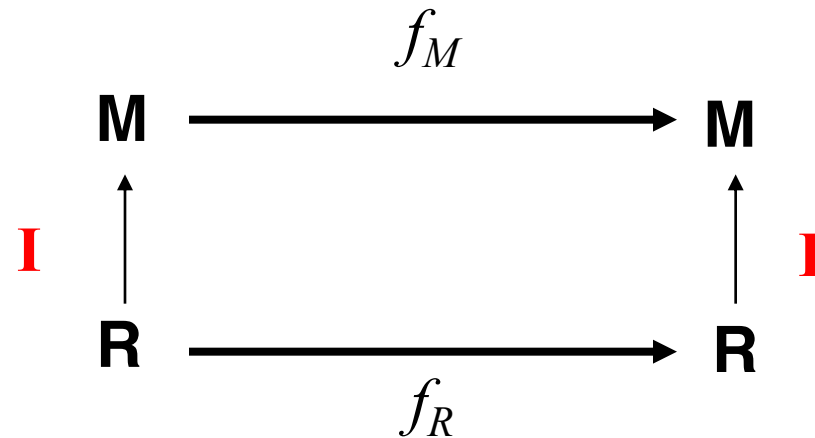
Grady Booch

Reality and Model

- Reality R: Real Things, People, Processes, Relationship

- Model M: Abstractions from (really existing or only thought of) things, people, processes and relationships between these abstractions

What is a "good" model?

$$f_M$$

$$\mathbf{M} \longrightarrow \mathbf{M}$$

$$\mathbf{I} \qquad\qquad \mathbf{I}$$

$$\mathbf{R} \longrightarrow \mathbf{R}$$

$$f_R$$

- I is the mapping of real things in reality R to abstractions in the model M

  - also called **Interpretation**

- Relationships, which are valid in reality R, are also valid in model M

  - f_R: relationship between real things in R
  - f_M: relationship between abstractions in M

What are software models for

- Capture and precisely state requirements and domain knowledge
- Think about the design of a software system
- Generate usable work products
- Give a simplified view of complex systems
- Evaluate and simulate a complex system
- Generate potential configurations of systems
  - all consistent configurations should be possible
  - not always possible to represent all constraints in the model (model is an abstraction !)

Modelling issues

- ## Coherence
  - different views of the system must be coherent

- ## Variations in interpretation and ambiguity
  - define where different interpretations of the model are acceptable

What should we model in RE?

- The objects and people that are of interest for the given problem
  - E.g., the aircraft and the sensors and actuators relevant to the braking system
- The relevant phenomena
  - Weels_turning, Reverse_enabled, …
- The goals, requirements, and domain assumptions

Which tools can we use for modeling?

- Natural language (e.g., Italian, English, …)
  - Pros: simplicity of use
  - Cons:
    - high level of ambiguity,
    - it is easy to forget to include relevant information
- A formal language (e.g., first order logic, Alloy, Z, …)
  - Pros:
    - possibility to use some tool to support analysis and validation
    - the approach forces the user in specifying all relevant details
  - Cons: you need to be expert in the use of the language

Which tools can we use for modeling?

- A semi-formal language like UML
  - Pros:
    - simpler than a formal language
    - imposes some kind of structure in the models
  - Cons:
    - not amenable for automated analysis
    - some level of ambiguity
- A mixed approach
  - Use a semi-formal language for the basics
  - Comment and complement the semi-formal models with explanatory informal text
  - Use a formal language for the most critical parts

Introduction