System Modelling

Course leader: Dr. PATARICZA András

Lecturer: HUSZERL Gábor

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group

The Course

- Course (VIMIAA00)
  - 3 in 1 (in Hungarian/English/German)
  - students of B.Sc. in computer engineering (2nd semester)
- Dr. PATARICZA András
  - professor in charge
- GÖNCZY László, BERGMANN Gábor
  - operative lead, organising
- HUSZERL Gábor (huszerl@mit.bme.hu)
  - English speaking lectures, …

The English Speaking Course

- Lectures and lead
  - HUSZERL Gábor
- Hand-on Trainings
  - HUSZERL Gábor

- Web site

  https://inf.mit.bme.hu/en/edu/courses/remo-en
  - Infos, news, slides, …

The Course

- 14 lectures
  - Thursdays 10-12 AM, I.B413
- 6 hand-on trainings
  - Fridays 10-12 AM (on odd Weeks), I.B145
  - First time on the 24th February
  - tests (<u>not</u> an entry test!)
  - presence mandatory
- 1 home assignment (with oral defence)
- 2 mid-term exams
  - with entry tests

Grading

- Mid-term exams (with entry tests): 35% + 35%
- Home assignment (with defence): 30% of the mark
  - all three of them with at least 40% of the points
  - 1 mid-term exam and the home assignment can be repeated

- Optional additional points:
  - Hand-on training tests (min. 5 tests out of 6): +5%
  - optional assignments for additional points

Home Assignment

- Mandatory submission, mandatory oral defence
- Personal Assignment
- Deadlines
  - Release of assignments: 3. semester week
  - First submission (opt.): 5. semester week
  - Final submission: 12. semester week
- Electronic submission
- Oral defence

Course Topics

- Visual data analysis
- Structure modelling
- Behaviour modelling
  - State based, process based
- Development of models
- Checking of models
- Performance modelling
- Simulation
- Benchmarking, Code generation

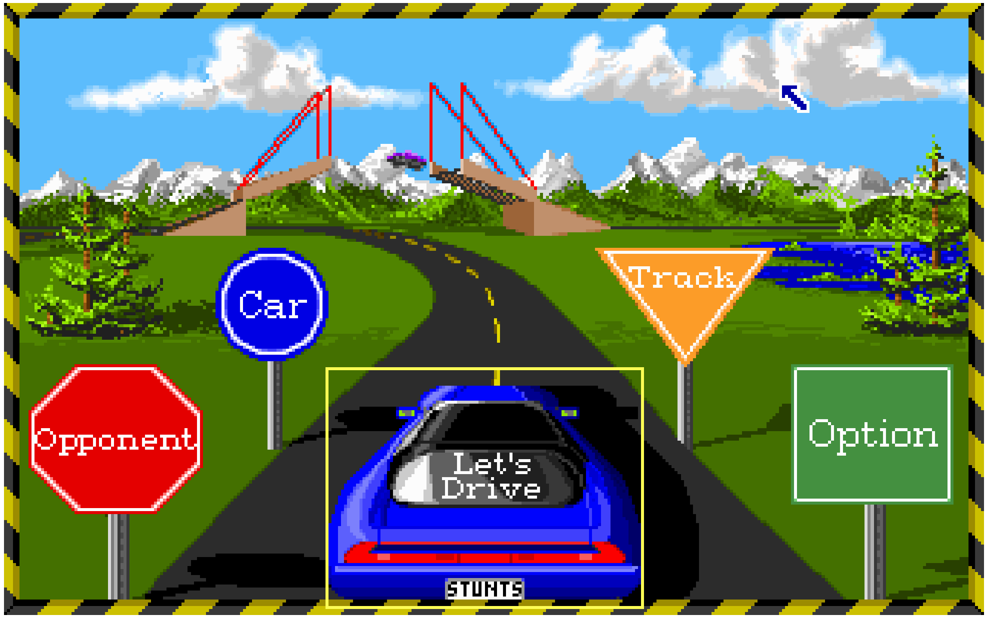
Foundations of Modelling

Dr. PATARICZA András, BERGMANN Gábor

HUSZERL Gábor

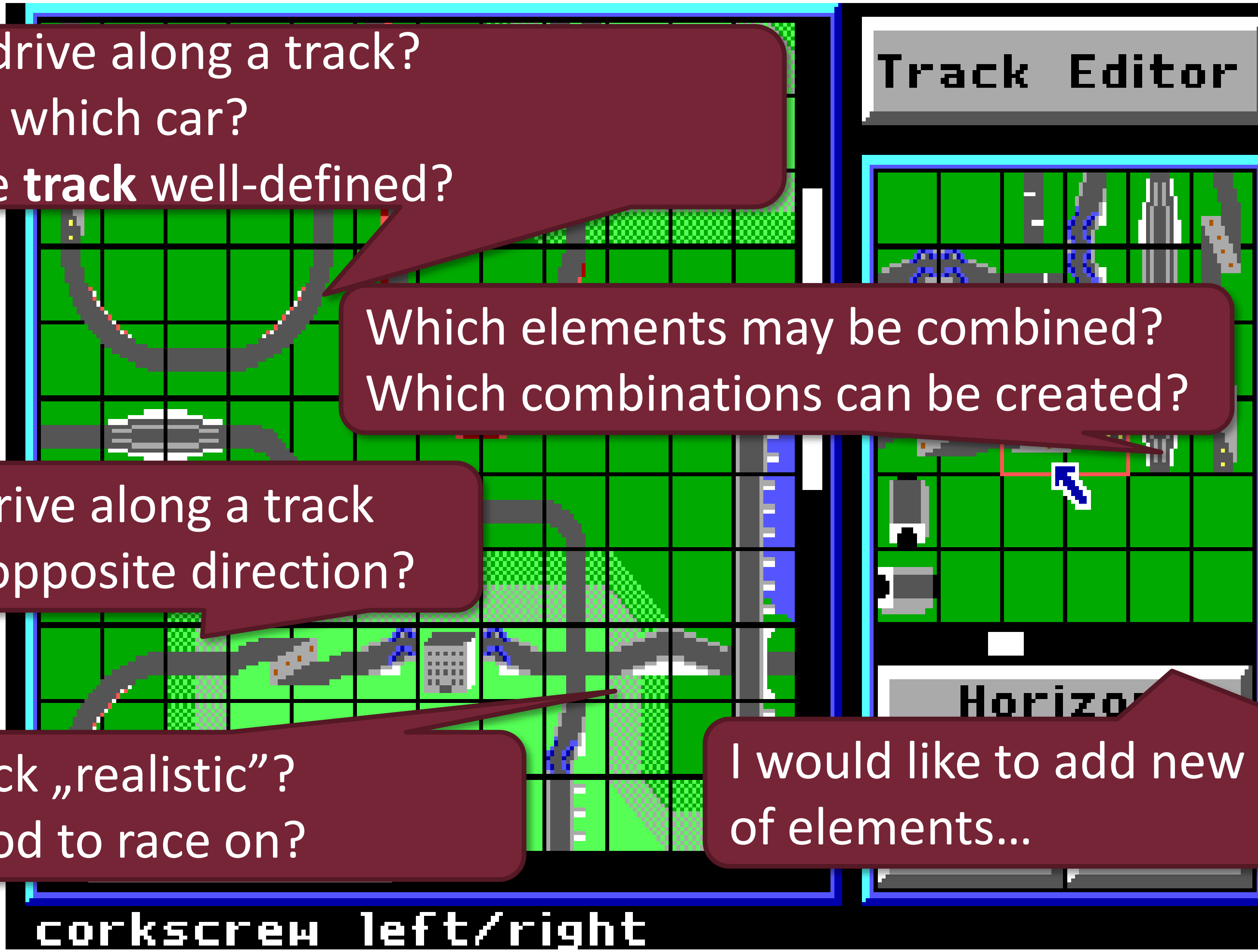
Budapest University of Technology and Economics
Fault Tolerant Systems Research Group

“Motivation”



- Stunts racing game, 1990.
  - Distinctive Games/Brøderbund Software
  - https://en.wikipedia.org/wiki/Stunts_(video_game)
- Racing + track creation (!)
  - „domain specific model”

“Motivation”: why model?



Can drive along a track?
With which car?
Is the **track** well-defined?

Which elements may be combined?
Which combinations can be created?

Can I drive along a track
in the opposite direction?

Is a track „realistic”?
Is it good to race on?

I would like to add new type
of elements…

Pic: http://www.abandonia.com/games/73/Stunts.htm

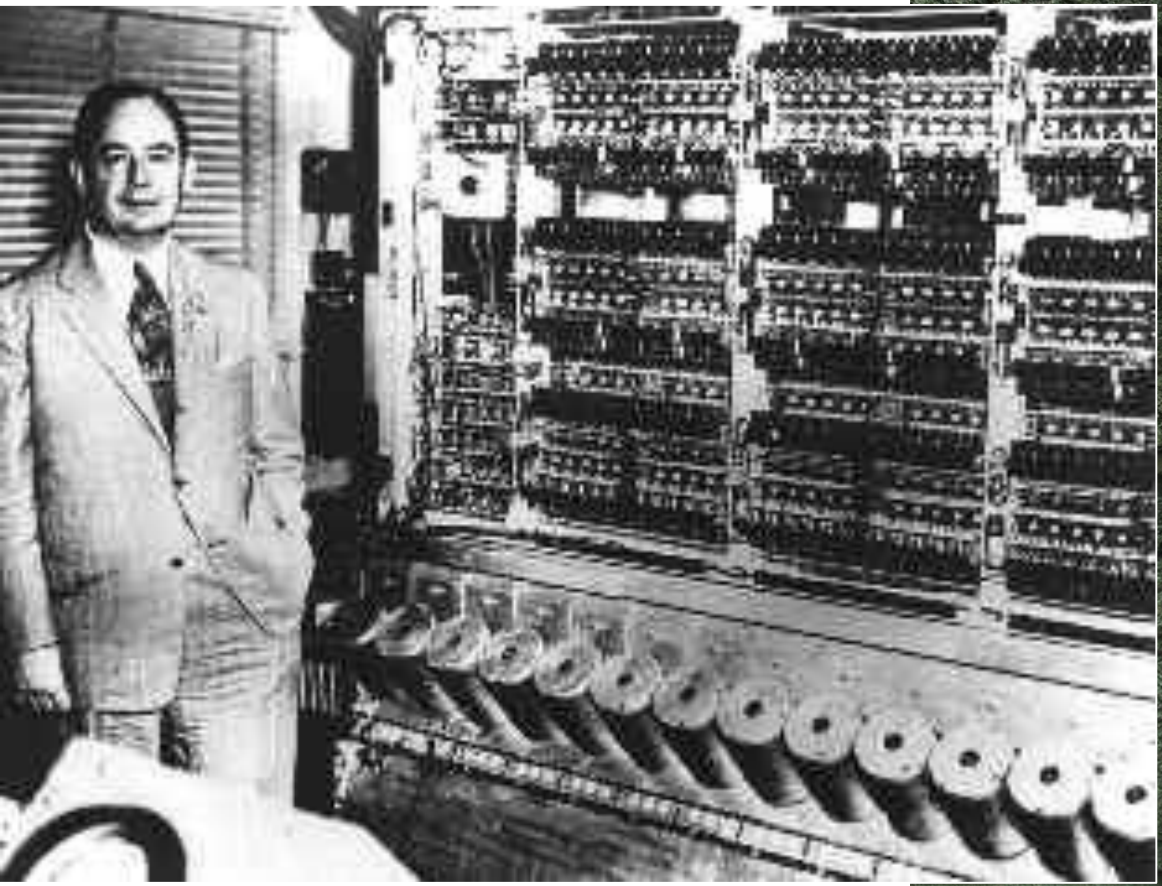
Table of Contents

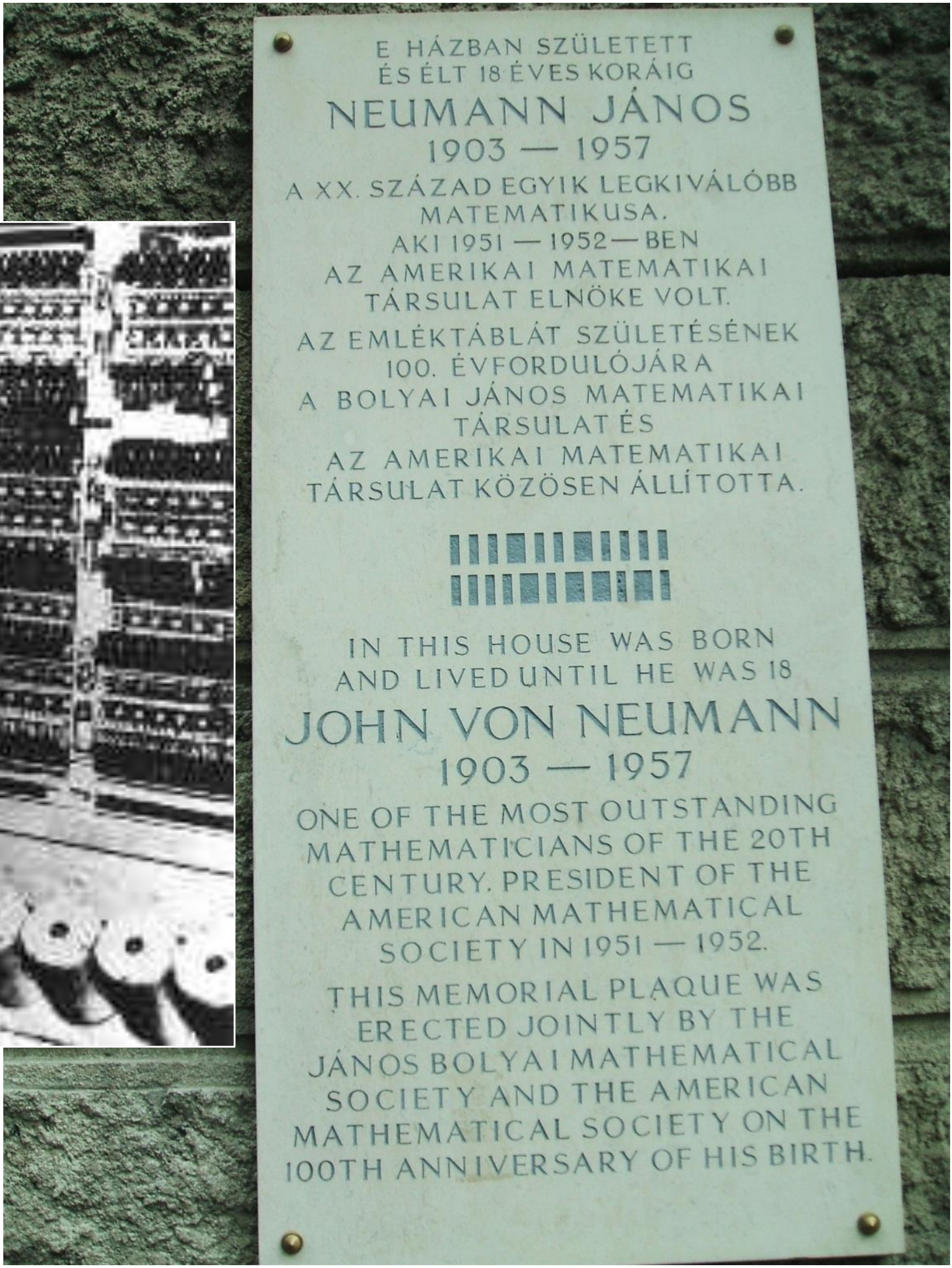
MODELS AND MODELLING

What is a model?

- "The sciences do not try to explain, they hardly even try to interpret, they mainly make models.
- By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena.
- The justification of such a mathematical construct is solely and precisely that it is expected to work – that is correctly to describe phenomena from a reasonably wide area.
- Furthermore, it must satisfy certain esthetic criteria – that is, in relation to how much it describes, it must be rather simple.„

John von Neumann




E HÁZBAN SZÜLETETT
ÉS ÉLT 18 ÉVES KORÁIG
NEUMANN JÁNOS
1903 — 1957
A XX. SZÁZAD EGYIK LEGKIVÁLÓBB
MATEMATIKUSA,
AKI 1951 — 1952 — BEN
AZ AMERIKAI MATEMATIKAI
TÁRSULAT ELNÖKE VOLT.
AZ EMLÉKTÁBLÁT SZÜLETÉSÉNEK
100. ÉVFORDULÓJÁRA
A BOLYAI JÁNOS MATEMATIKAI
TÁRSULAT ÉS
AZ AMERIKAI MATEMATIKAI
TÁRSULAT KÖZÖSEN ÁLLÍTOTTA.
IN THIS HOUSE WAS BORN
AND LIVED UNTIL HE WAS 18
JOHN VON NEUMANN
1903 — 1957
ONE OF THE MOST OUTSTANDING
MATHEMATICIANS OF THE 20TH
CENTURY, PRESIDENT OF THE
AMERICAN MATHEMATICAL
SOCIETY IN 1951 — 1952.
THIS MEMORIAL PLAQUE WAS
ERECTED JOINTLY BY THE
JÁNOS BOLYAI MATHEMATICAL
SOCIETY AND THE AMERICAN
MATHEMATICAL SOCIETY ON THE
100TH ANNIVERSARY OF HIS BIRTH.

What is a model?

- Model is a **partial and simplified depiction of the real or a hipothetical world (the „system”)**, of which can be **substituted** for specifc viewpoints
- Decisions:
  - **Which part of the world**?
  - **What is simplified**?
  - **How it can be mapped to the world**?

When is it possible or recommended to use?

- Benefits
  - smaller (finite)
  - has managable size

What is NOT a model?

- The model is NOT the reality!

- The model is not a diagram.
  - It is only a view…


Ceci n'est pas une pipe

Model vs. Real World

Matematical Models vs. Real World

- All of the models are: closed world
  - Effects, factors
  - Parameters
  - Validity
- The models have no defined behavior outside its scope
- Not all aspects can be expressed in advance
  - *Human decisions*
  - *Generate models*
- *Validation of the solutions*
- *Constructed to answer questions…*

- Normal operations
  - Scope:
    - Enough materials are available
    - All of the orders can be completed in time
  - Target function:
    - Minimal cost
- Corner case
  - Scope:
    - Not enough materials are available
  - Target function :
    1. As many as possible orders completed in time
    2. Minimal cost

Example: safety critical SW

- Braking of an airplane: wheel brake + thrust

1993 Warsaw: Lufthansa 2904

- (SW) protection:

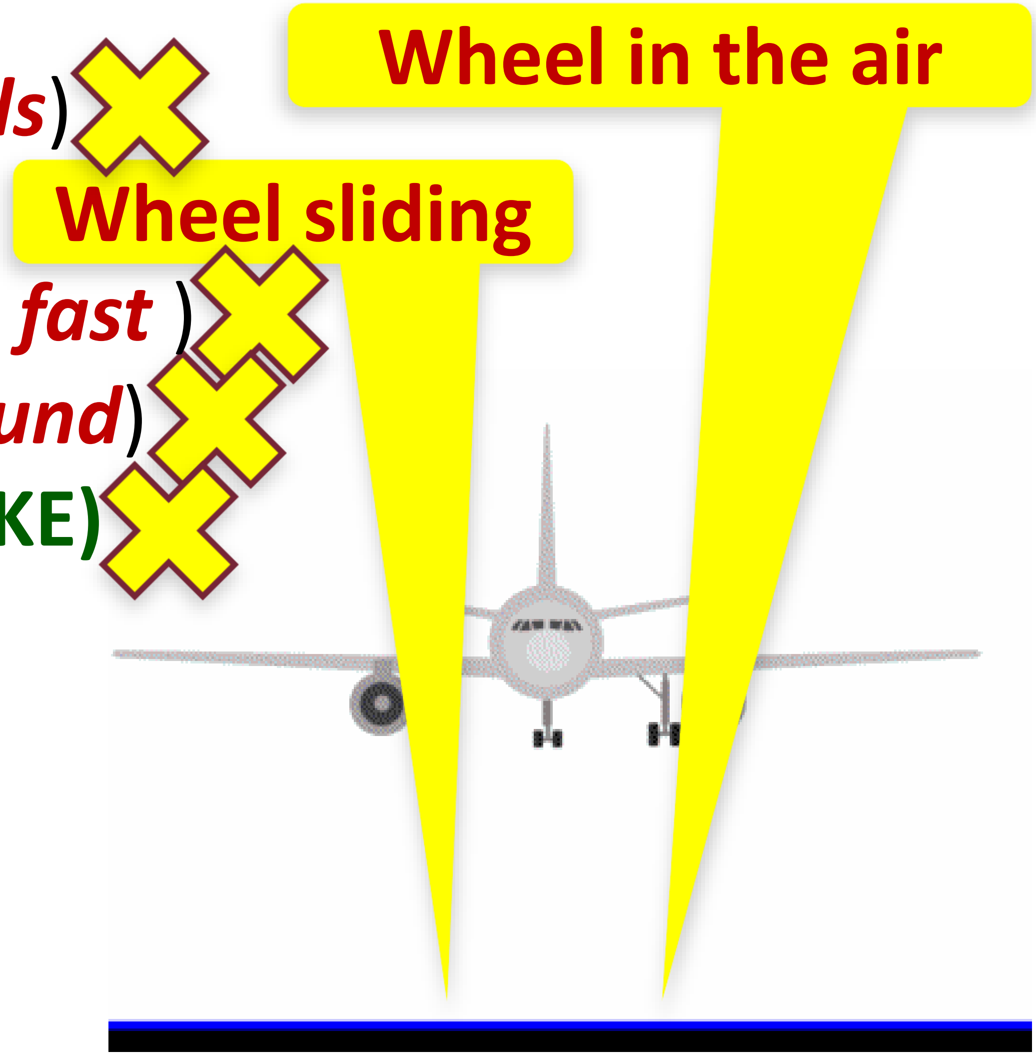
(***load on both weels***)

OR

(***one wheel is rotating fast***)

→ (***plane on the ground***)

→ **(PILOT CAN BRAKE)**



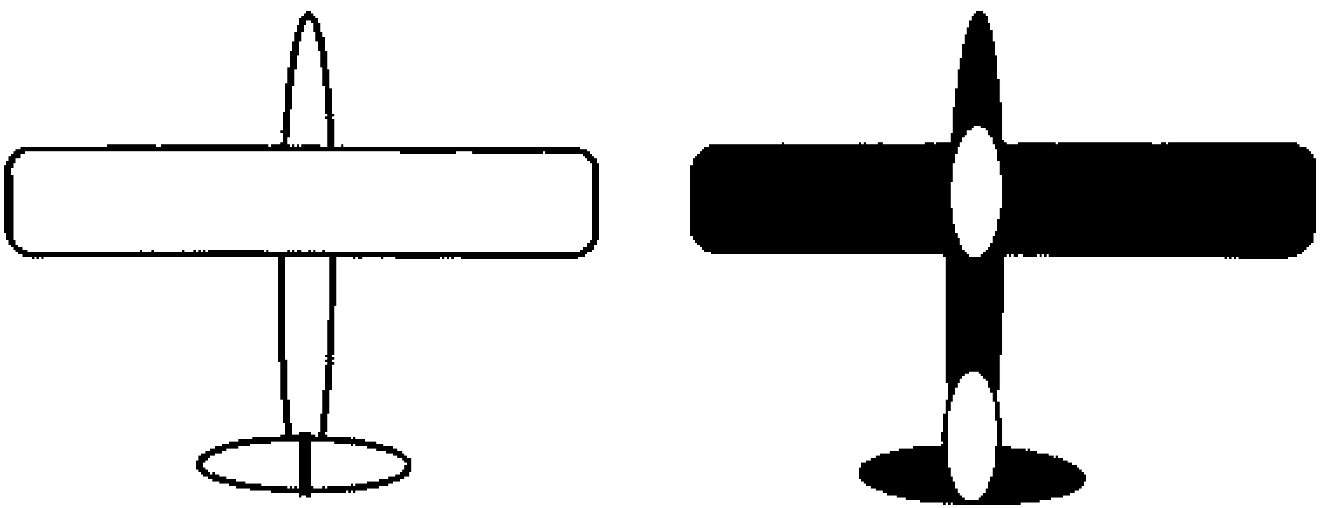
Model quality

- Reality: open world ↔ **Model:** closed world

- "Faithfullness" of models:
  - Probable + critical cases

- Implementation of a bad model can be deadly…

Armor? → Abraham Wald



Armor? → Abraham Wald

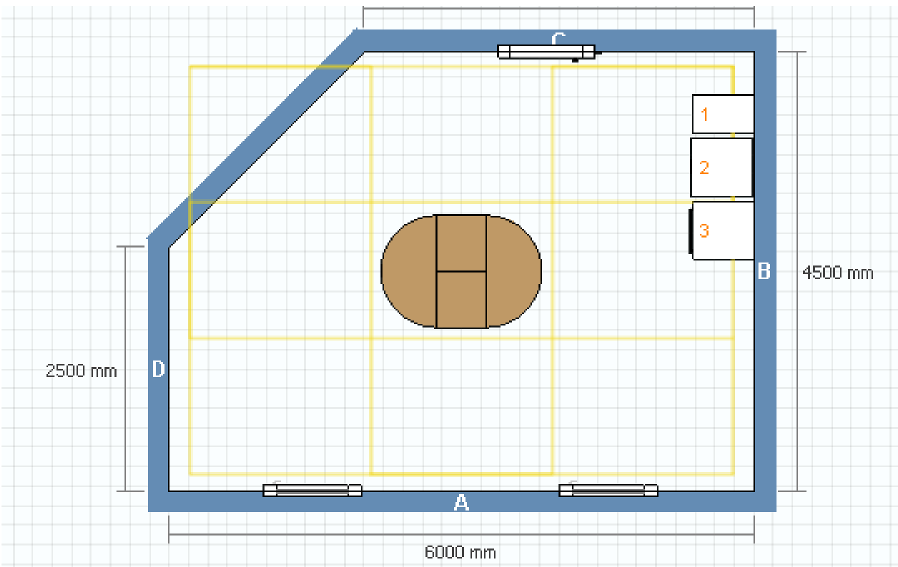
Model is valid only with environment...

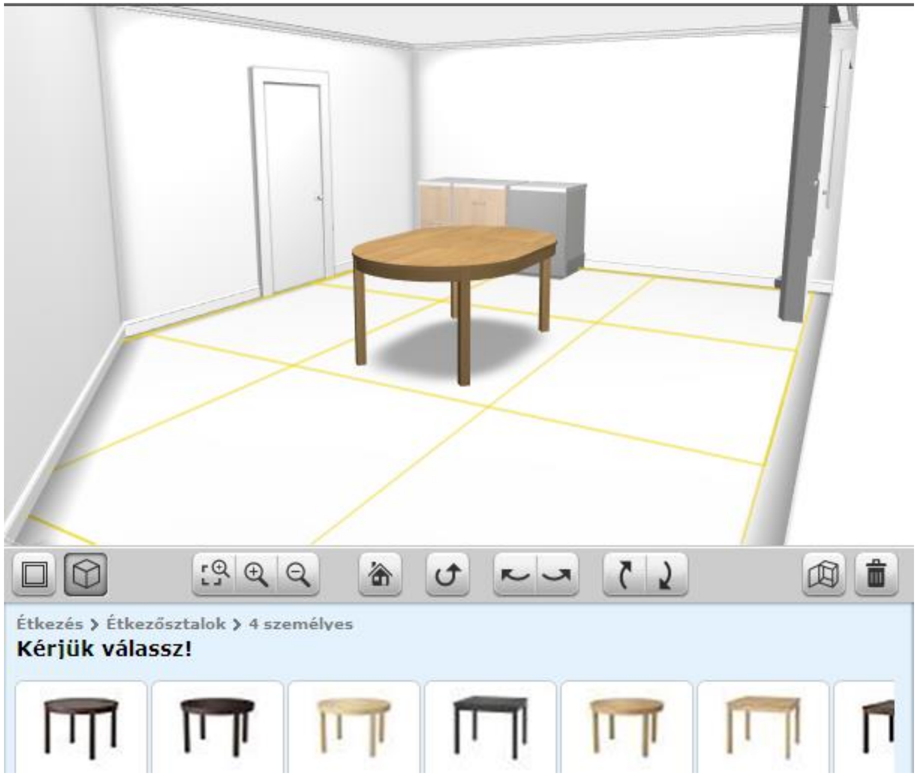
What is the point of modelling?

- I am developing software. Should I do modelling?
  - Yes! In fact, you already do!
    - (The source code is a kind of model…)
    - Most importantly: **mental models**
    - When should be these models documented?
  - Role of the model: **communication**
    - Human → Human
    - Human → Machine
    - Machine → Machine
    - Human → Human (themselves in the future)
      - E.g. Why we implemented this algorithm this way some years ago…

Modelling in real life?

E.g.: web-based kitchen designer
[of a Scandinavian company]

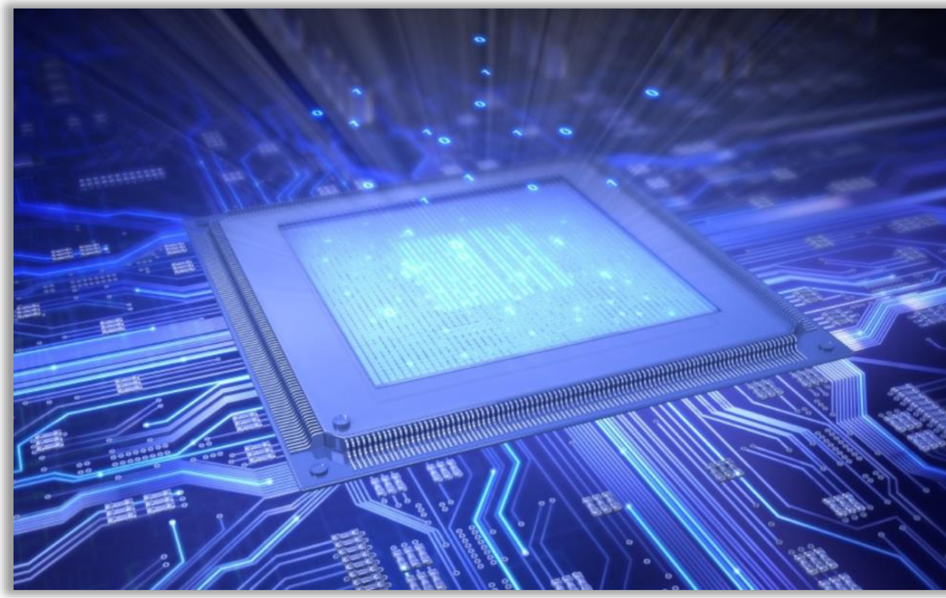




This is also a modelling language!

- VHDL, Verilog – domain specific hardware description

```
...
ARCHITECTURE Struct OF MyLogic IS

   COMPONENT And2 IS
      PORT (x, y: IN std_logic;
            f: OUT std_logic);
   END COMPONENT;

   COMPONENT CustomHW IS
      PORT (x: IN std_logic;
            f: OUT std_logic);
   END COMPONENT;

   SIGNAL n1, n2: std_logic;

BEGIN
      And2_1: And2 PORT MAP (    ,     ,     );
      And2_2: And2 PORT MAP (    ,     ,     );
      CustHW: CustomHW PORT MAP (     ,     );
END Struct;

```



Modelling language

- The purpose of the model is to help the communication
  - Understanding of the model is necessary
  - **Modeling languages** (when do we need them?)
- Syntax
  - „Matematical structure”: abstract syntax
  - Notation: concrete syntax
    - graphical symbols / textual format
- Semantics
- Boundary conditions, constraints
  - Syntactical correctness, well-formedness
  - Design conventions (can vary by teams)

Models and modelling

Purpose of the models

Basic Concepts

Examples

PURPOSE OF THE MODELS

Models in Engineering Jobs

- Engineering Jobs
  (Architects, Mechanical Eng., Civil Eng., Electro Eng., ...,
  Landscape Architects, ...)

It is the widely used engineering approach: **Planning**

System Design Process

ANSI/EIA 632 standard

Management
- purchasing, supplies
- Planning, operations, assessment

Engineering tasks

System Design
- Requirement definitions
- Solution definitions

Product Development
- Implementation
- Usability

Quality Check
- System Analysis
- Requirements validation
- System Verification
- End-product Verification

These steps are ususally supported by models

System Design Process - Analogy

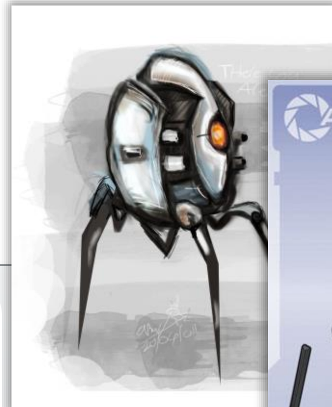
PORTAL 2

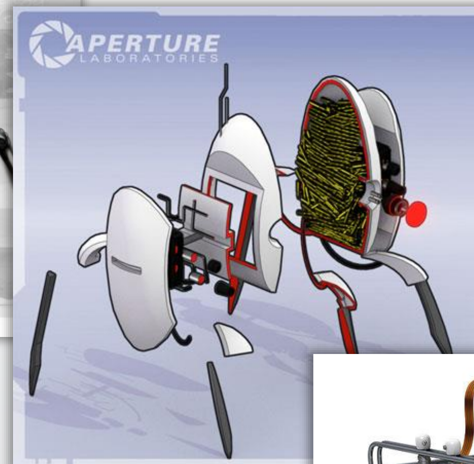
Management

System Design

Product Development

Quality Check





Engineering Tasks

Validation: Are we building the right product?
Verification: Are we building the product right?

Application – Documentation

- The model is simpler to handle
  - Easier to communicate than the whole reality
  - Can be refined step-wise (explained later)
- Communication, Visualization
  - Demonstrations (explained later)
  - Easy to understand textual language
  - Expressive diagram
- Supporting conceptualization, design
  - Similar viewpoints
  - „thinking tool”

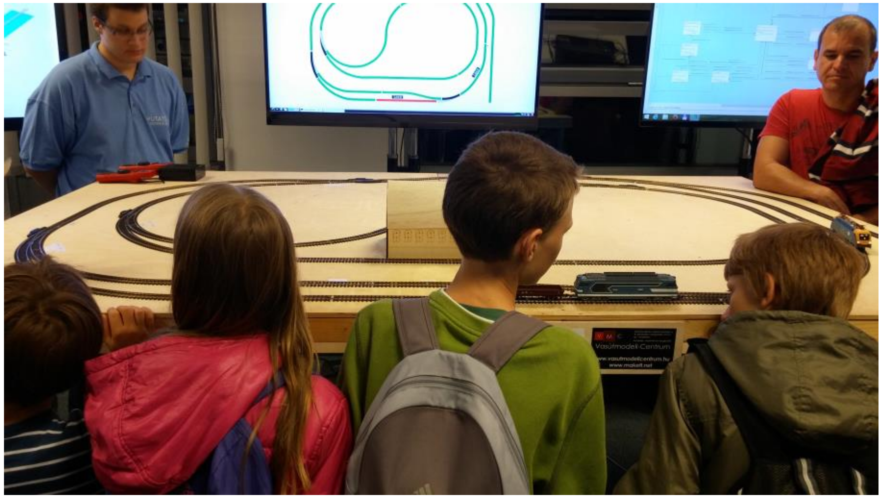
Application - Analysis

- Human contribution or (semi-)automated
- Method
  - Review, static analysis
  - Dynamic state space exploration – model checking
  - Proving theorems about the model
- Goal
  - Detecting errors (best effort)
  - Proving safety properties (stronger reqirement!)
  - Calculating properties of the model (e.g. timing)

Application – Deriving Artifacts

- Human contribution or (semi-)automated
- Output
  - Generating program code, analyzable language, etc.
  - Another model
    - Refinement, next design (abstraction) phase
    - Partial view
    - Model integration
  - May preserve model properties

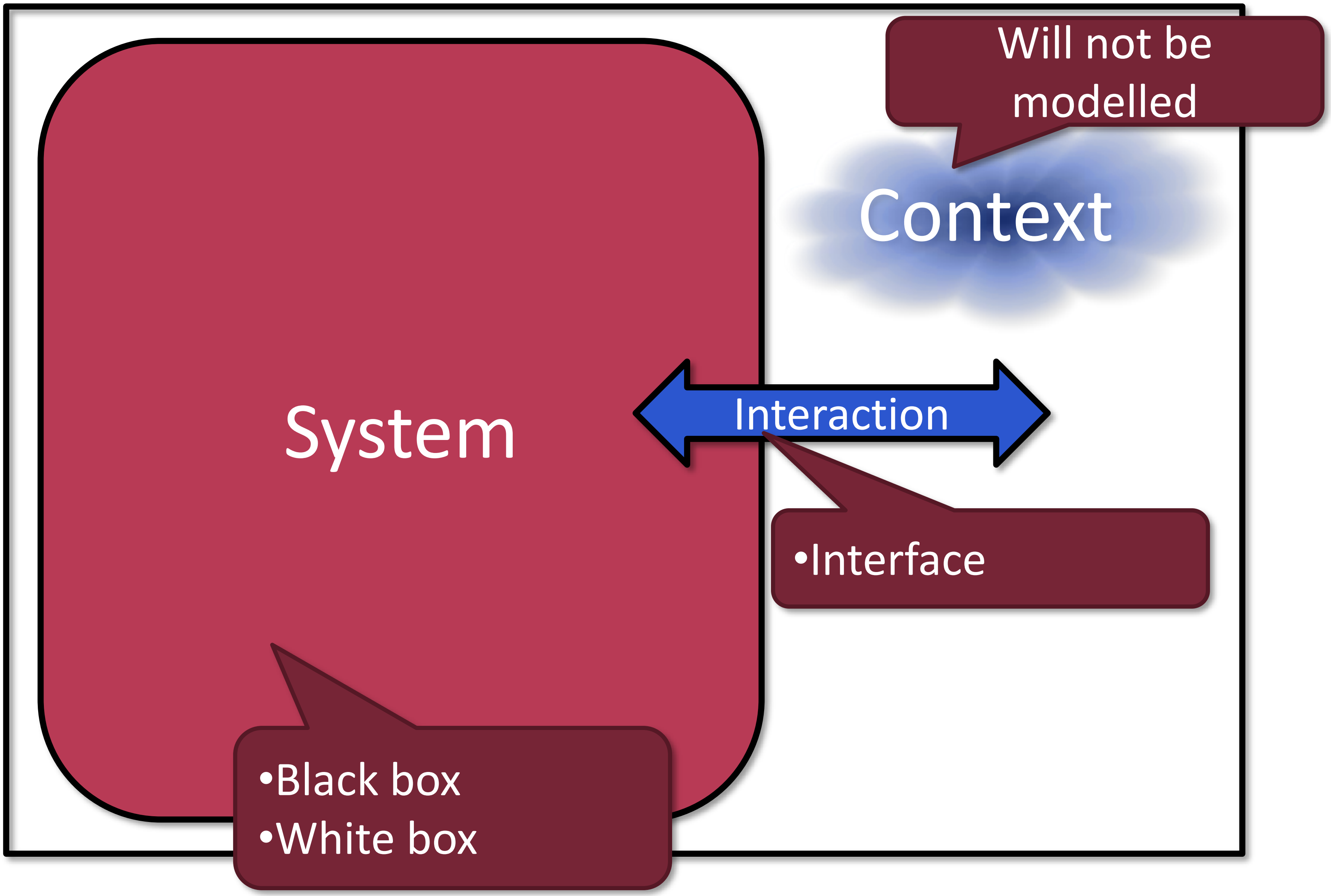
Usage of models - Simulation



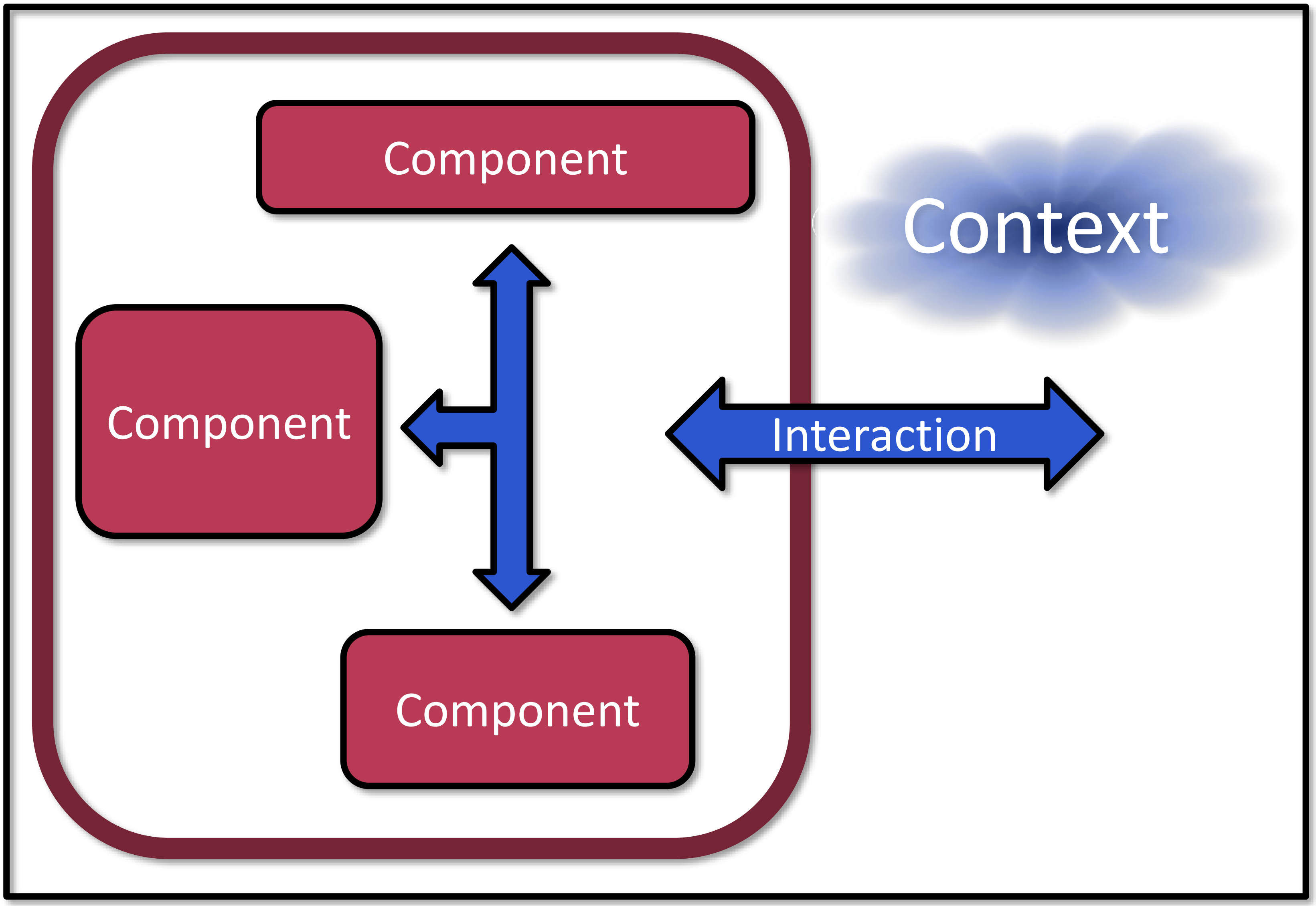
- Demonstration
  - A tool for communication
- Validation
  - „I built it correctly… but did I build the right thing?”
- Experiment
  - Proof for desired system characteristics
  - Measurement of quantitative properties
  - Substitute expensive real-life experiments
  - Properties that cannot be predicted using formulae

BASIC CONCEPTS

Basic concepts – System and Context



Basic concepts – System and Context

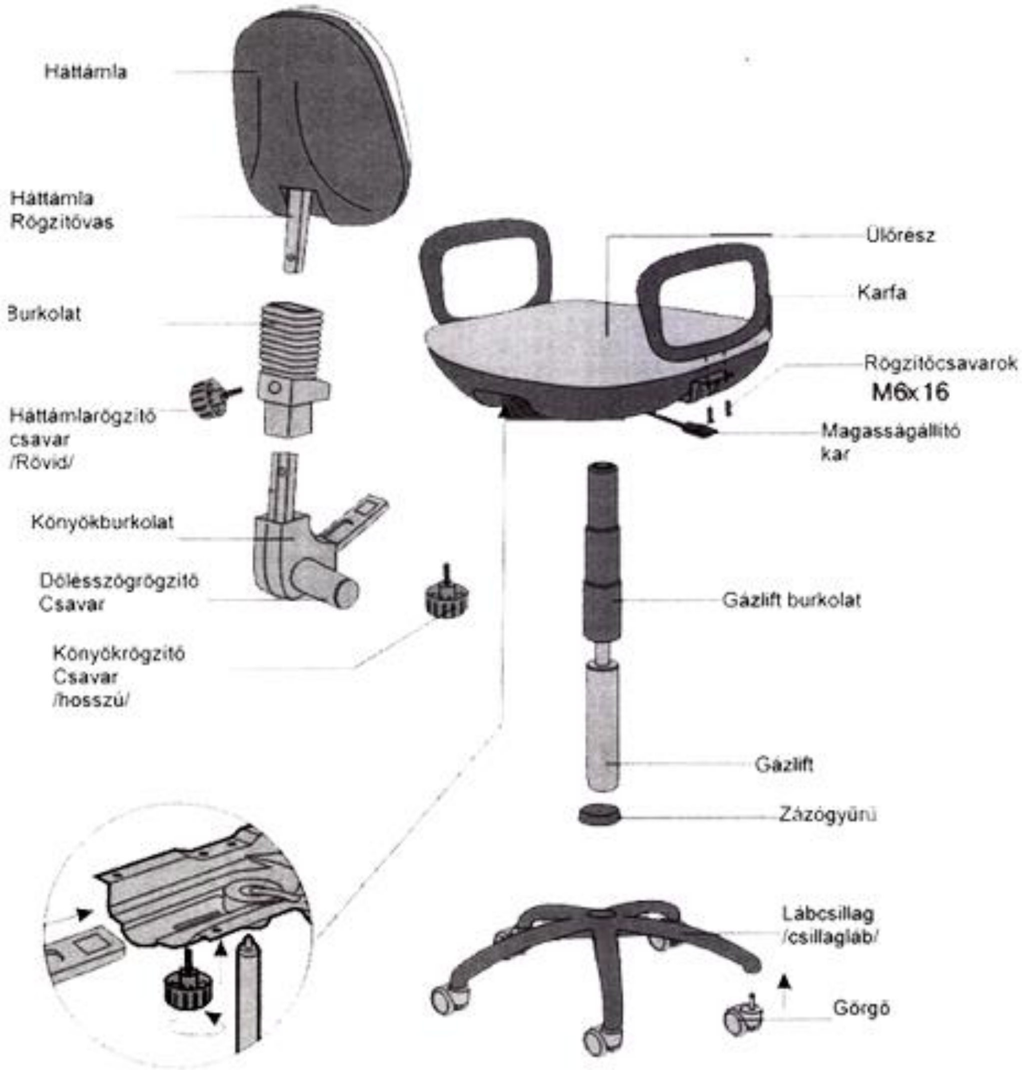


Basic Concepts – Refinement/Abstraction

- *Refinement*: adding further details to the model…
- …keep the abstraction of the input model
- Inverse: *(vertical) abstraction*
- There was a *hierarchial refinement* on the previous slide
  - „refinement of the boxes”
- Other aspects can be refined as well...
  - E.g. partitioning of a set (variable domain refinement)
    - **Good** / **bad** can be refined with:
    - **Fast** / **average** / **slow** / **incomplete** / **dangerous**

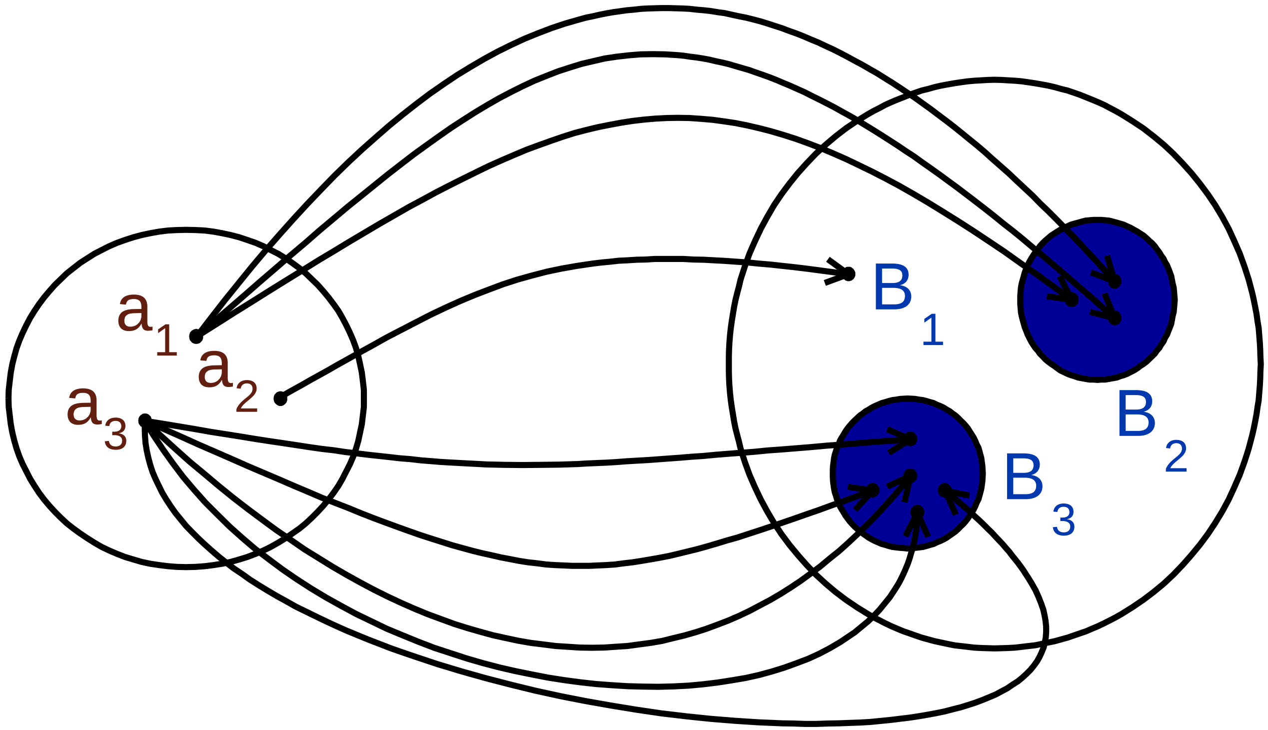
Basic Concepts - Refinement

Basic Concepts - Refinement



Partitions of Set Elements

Mapping of disjunct subsets to elements of a set



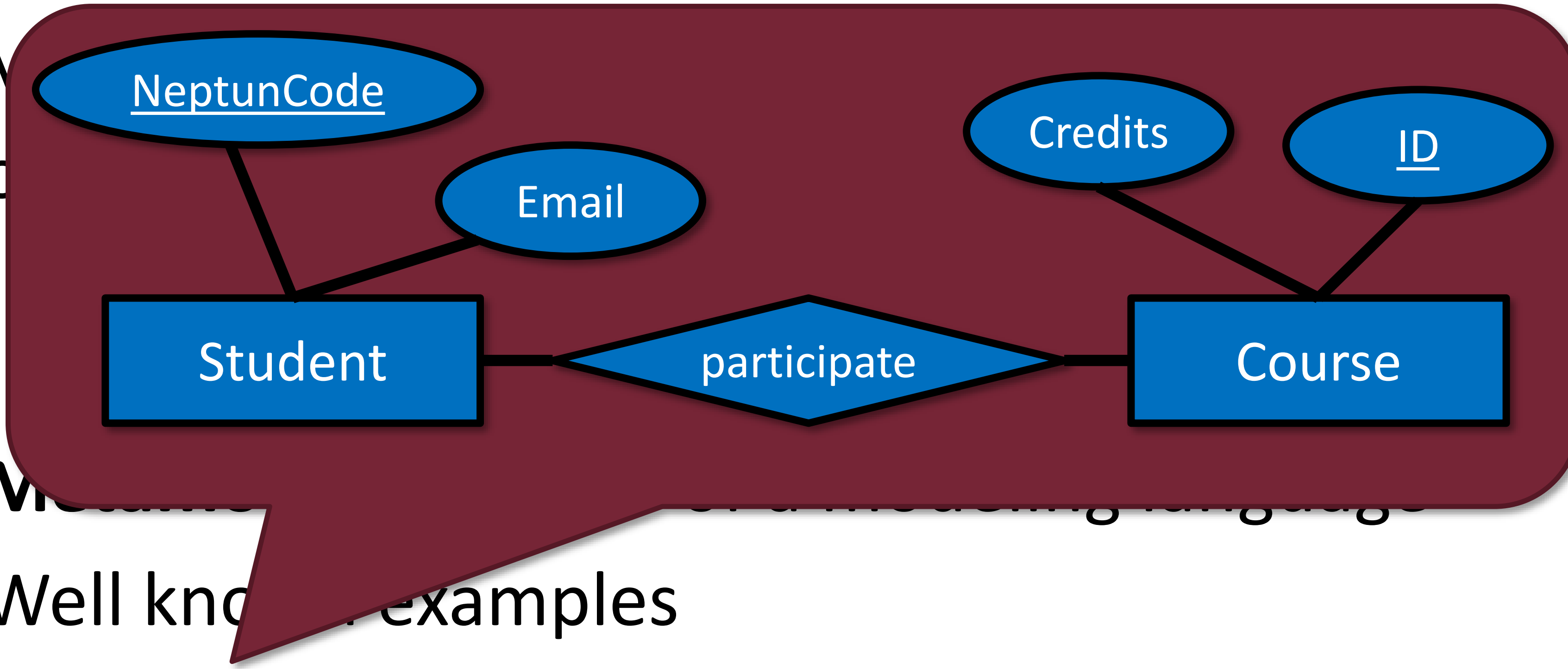
$$\forall a_i, \in A, R(a_i) \subset B \mid \forall a_i, a_j\ R(a_i) \cap R(a_j) = \varnothing$$

Basic Concepts - Metamodelling

- Modeling language: what are the types of the objects?
  - + what can be the connection between the objects?
  - + how the types can be related to eachother?
- **Metamodell** = model of a modeling language
- Well known examples
  - Entity-Relationship (ER) model
  - UML object diagram $\rightarrow$ class diagram
  - Database table $\rightarrow$ Relational database schema
  - XML document $\rightarrow$ XML schema (or DTD)
  - …

Basic Concepts - Metamodeling

- N
  c



- **M**
- Well kn examples
  - Entity-Relationship (ER) model
  - UML object diagram → class diagram
  - Database table → Relational database schema
  - XML document → XML schema (or DTD)
  - …

Model and modelling

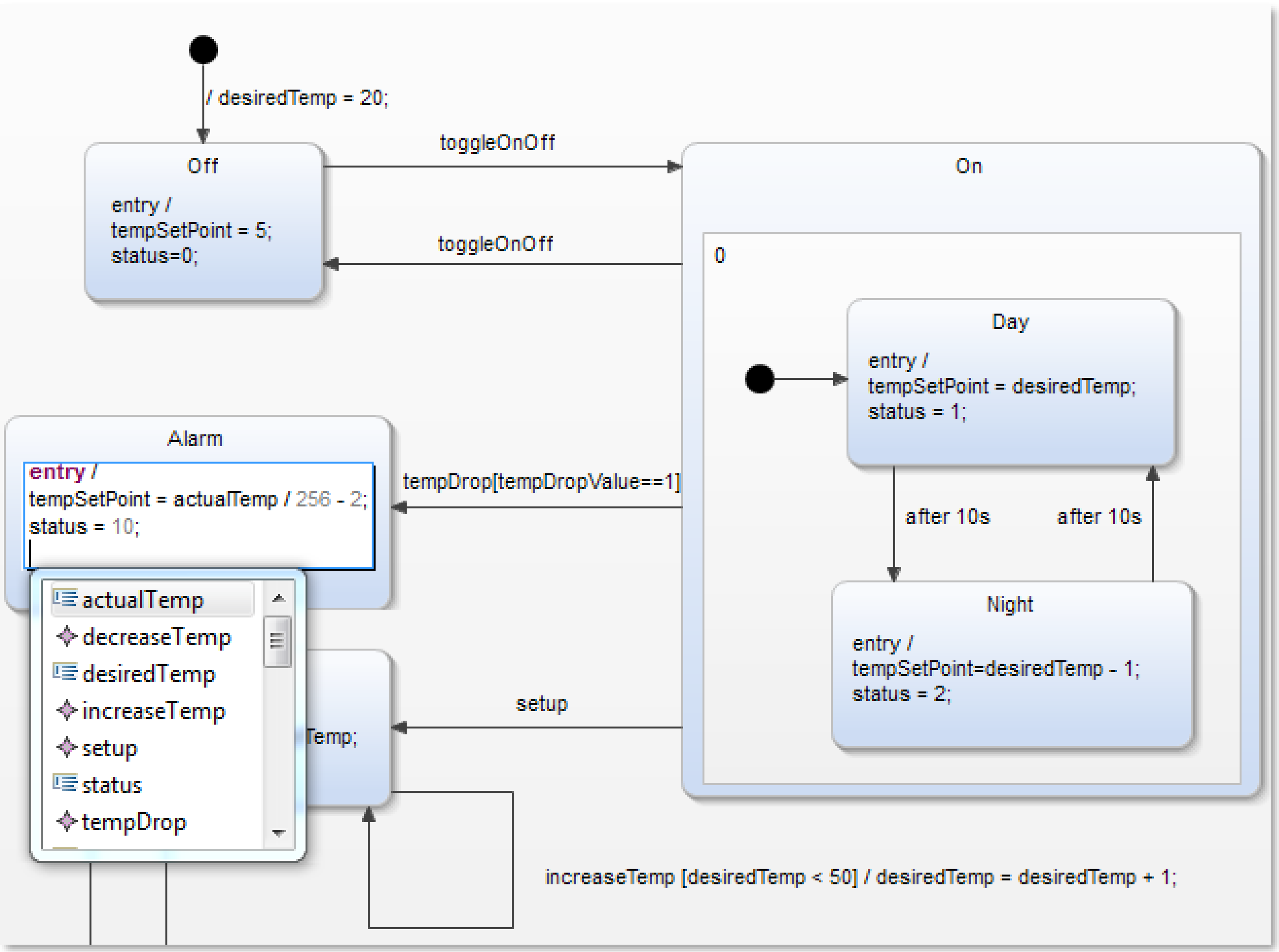
Purpose of the models

Basic Concepts

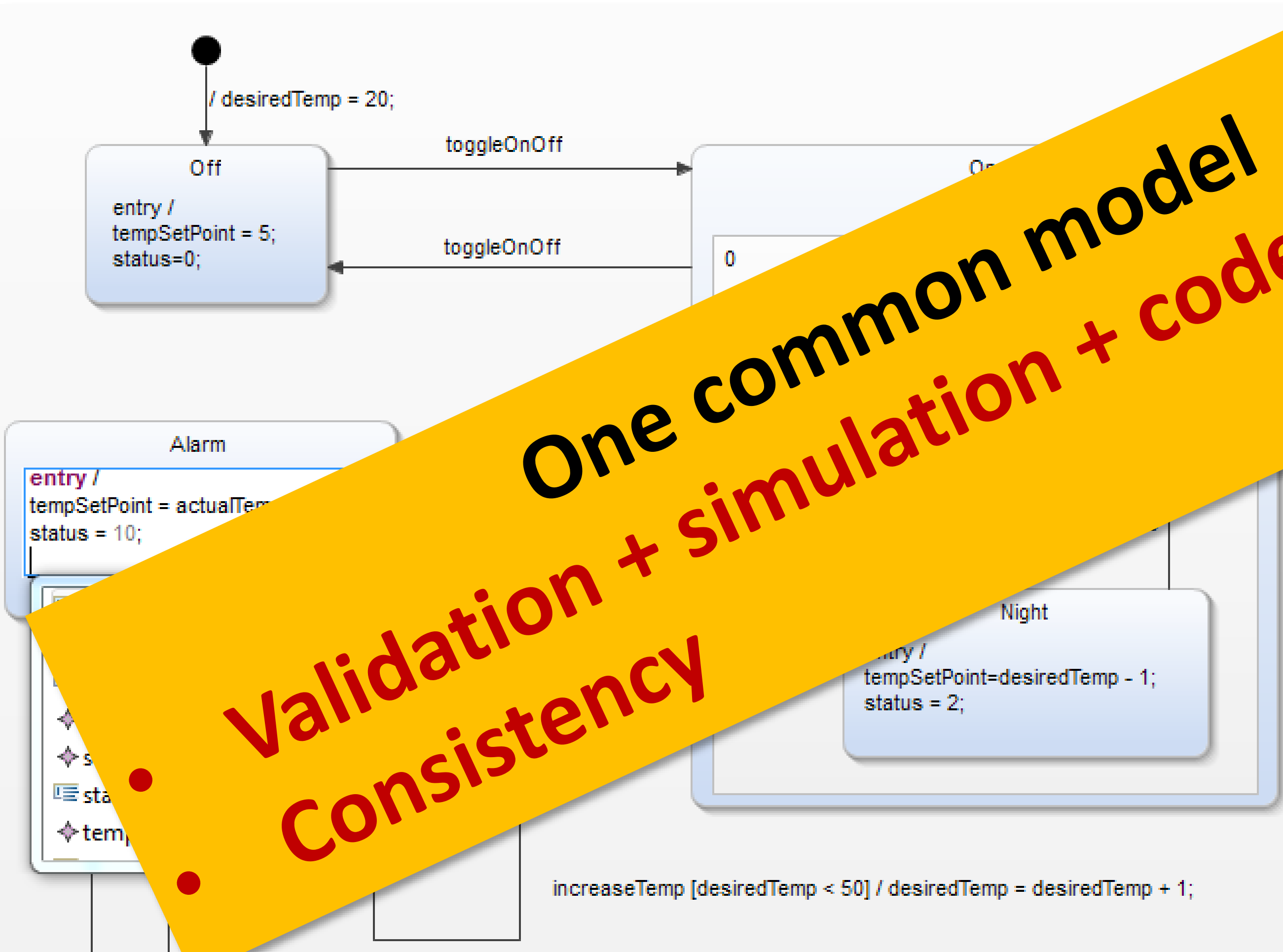
Examples

EXAMPLES

Yakindu – State Charts



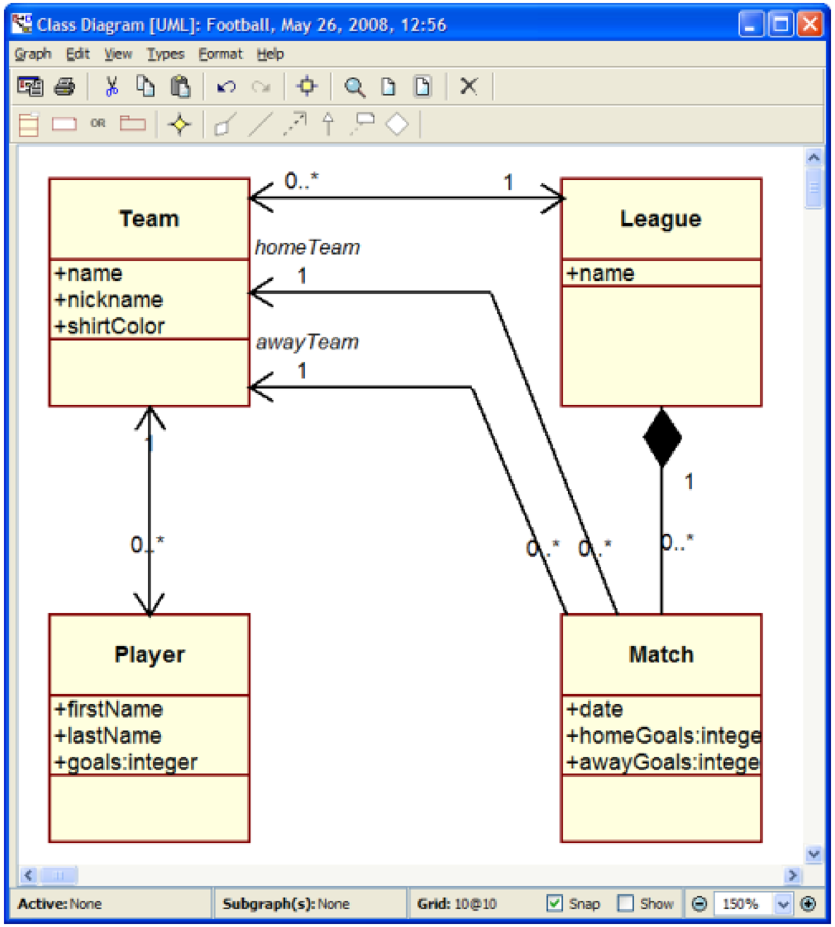
Yakindu – State Charts

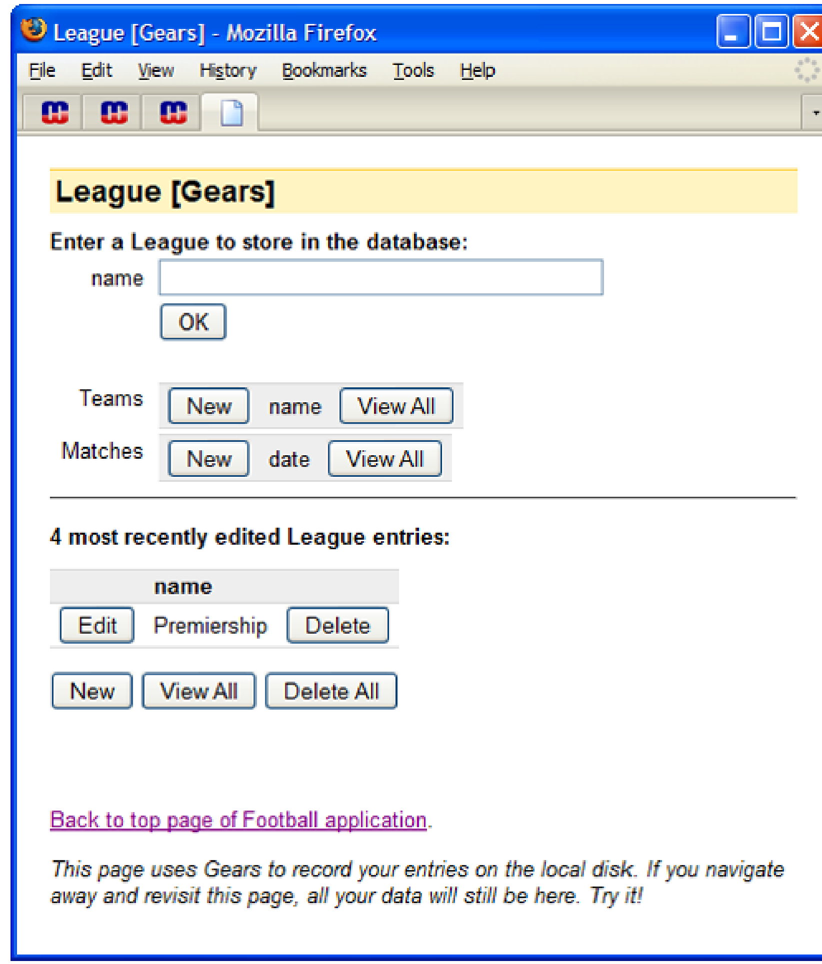


One common model

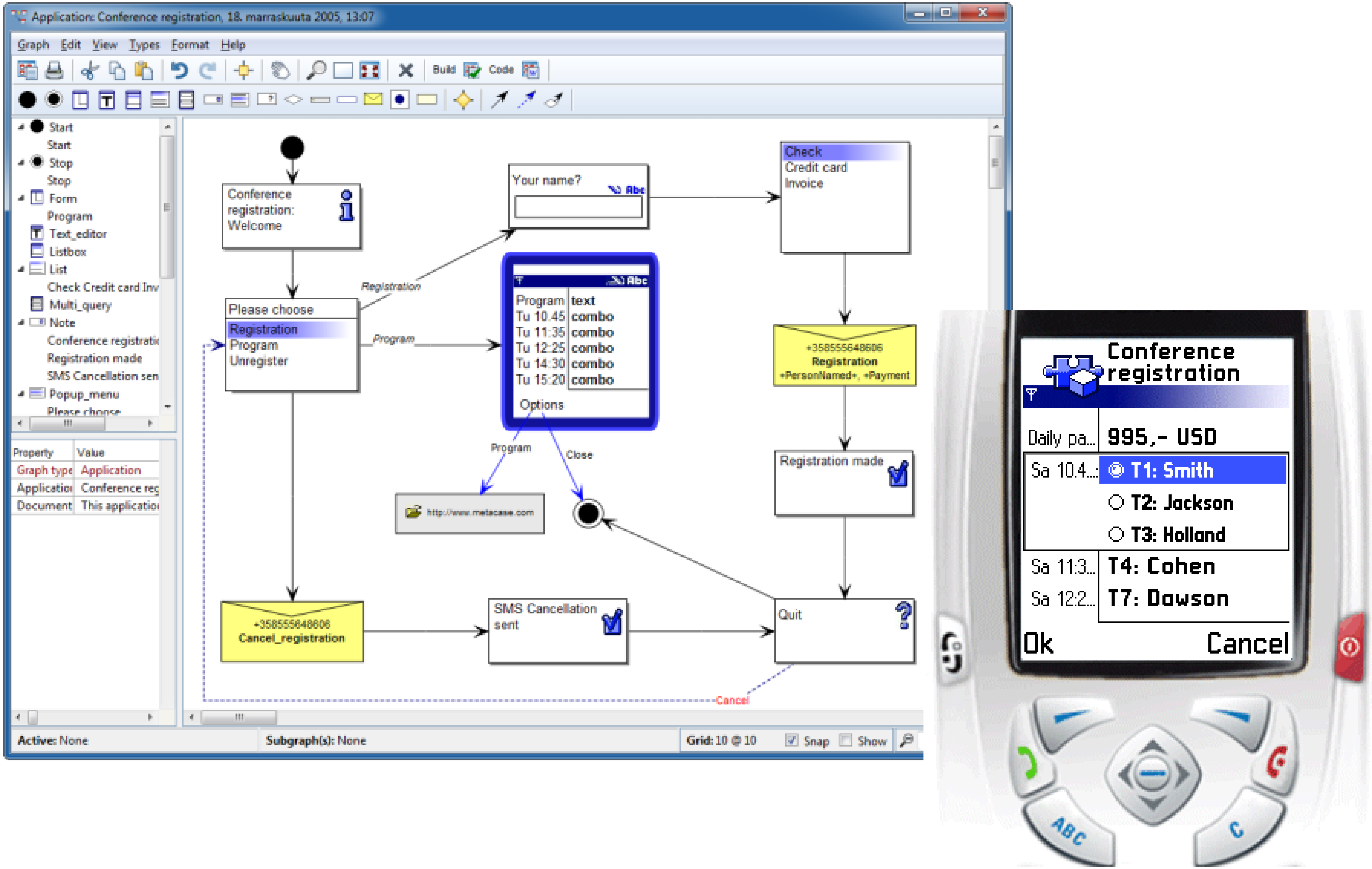
- Validation + simulation + code generation
- Consistency

Web Application Development





Smartphone App Development



Some representative projects…

R3COP (EU ARTEMIS)

- Automated testing of robots
- Robostusness and securtiy analysis
- ARTEMIS Innovation Award 2012
- Altogether: 15 EU projects

TRANS-IMA (Embraer)

- Eclipse based development tooling
- HW-SW allocation: avionics architecture
- Integration to the distributed Embraer simulator
- (1st time in Europe)

Data Storage Systems (IBM)

- Supply-chain simulation and optimization
- Prediction of order data
- IBM Vác: Data Storage Systems
- Supply Chain Technology Award 2012

VCL: Virtual Computing Lab

- Open source cloud infrastructure
- Apache project
- Education: lab courses” (BYOD)
- First time in Hungary
- Tempus Award