ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Laboratory for Safety Analysis
INSTITUTE FOR ENERGY TECHNOLOGY

Reliability of Technical Systems

Advanced Methods for Systems Modeling and Simulation II: Object-oriented modeling and MC modeling

Main Topics

1. Short Introduction, Reliability Parameters: Failure Rate, Failure Probability, etc.
2. Some Important Reliability Distributions
3. Component Reliability
4. Introduction, Key Terms, Framing the Problem
5. System Reliability I: Reliability Block Diagram, Structure Analysis (Fault Trees), State Model.
6. System Reliability II: State Analysis (Markovian chains)
7. System Reliability III: Dependent Failure Analysis
8. Data Collection, Bayes Theorem, Static Redundancy
9. Combined Redundancy, Dynamic Redundancy; Advanced Methods for Systems Modeling and Simulation I: Petri Nets
10. Advanced Methods for Systems Modeling and Simulation II: Object-oriented modeling and MC modeling
11. Human Reliability Analysis
12. Software Reliability, Fault Tolerance
13. Case study: Building a Reliable System

Advanced Methods for Systems Modeling and Simulation

Object-oriented modeling for the reliability analysis of infrastructure systems

Part I: Short Introduction to Object-oriented Modeling

- Stochastic simulation – basic Monte Carlo methods for reliability analysis
- Object-oriented modeling approach – framework
- How to build an object-oriented model

Part II: Application to Complex Engineering Systems:

Reliability analysis of large-scale electric power systems

- The IEEE Reliability Test System 1996 and its implementation
- Results

Part I: Short Introduction to Object-oriented Modeling

Stochastic simulation – basic Monte Carlo methods

Simulation:	an abstraction of a real system by a computer program in order to mimic and analyze its behavior

Monte Carlo technique:	stochastic simulation using algorithmically generated random numbers

Monte Carlo methods – two simple examples

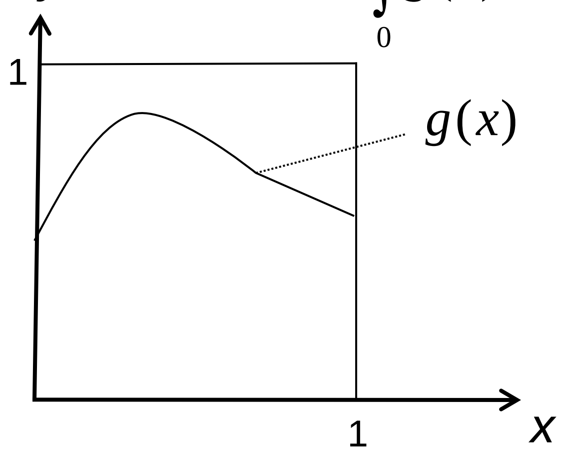
Example 1:

A fair die is thrown. What is the probability of a one occurring on the upper face? This is obviously 1/6 as each of the six faces has equal probabilities of occurring. This probability can be estimated by sampling simulations. Throw the die *N* times and record the times number one occurs. Let this be *f* times. The probability is *f/N*. As *N* increases sufficiently, *f/N* approaches 1/6.

Example 2:

How do you calculate the following integral by sampling simulations?

$$I = \int_0^1 g(x)dx$$



Basic Monte Carlo methods for reliability analysis

A simple example for estimating the unavailability Q of a system :

Assume a system consisting of k components, where:

s_i: state of the *ith* component (boolean)
Q_i: failure probability of the *ith* component
R_i: random number for the *ith* component; U_i ~uniform[0,1]

then, assuming independent failures :

$$s_i \begin{cases} 0 \quad \text{(success)} \quad \text{if} \quad U_i > Q_i \\ 1 \quad \text{(failed)} \quad if \quad 0 \le U_i \le Q_i \end{cases}$$

Basic Monte Carlo methods for reliability analysis (II)

1. sample the states of all components („throw the dices“) to get the system state *s*:

$$s = \{s_1, \ldots, s_i, \ldots, s_k\}$$

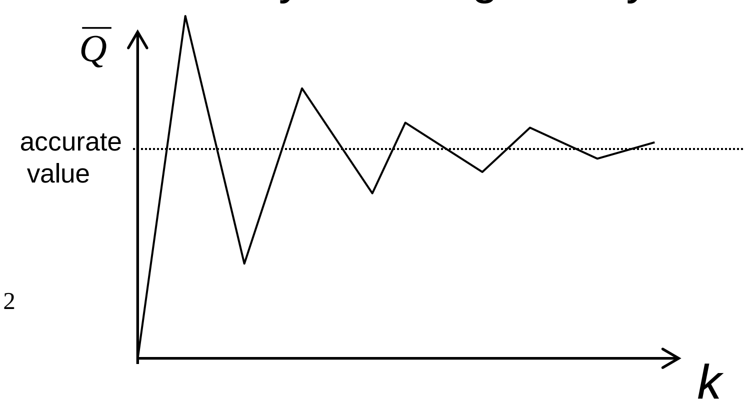
2. Perform system analysis to judge whether *s* is a failure state or not:

$x_j = 0$ if the system is in the up state

$x_j = 1$ if the system is in the down state

3. Performing *N* system state samples, the unbiased estimate of the system unavailability then is given by:

$$\overline{Q} = \frac{1}{N} \sum_{j=1}^{N} x_j$$

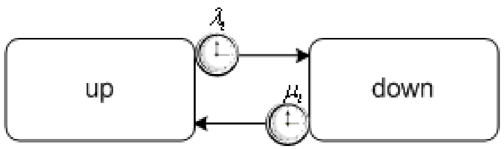


with variance:

$$V(\overline{Q}) = \frac{1}{N} V(x) = \frac{1}{N(N-1)} \sum_{j=1}^{N} (x_j - \overline{Q})^2$$

Stochastic simulation – sequential Monte-Carlo simulation

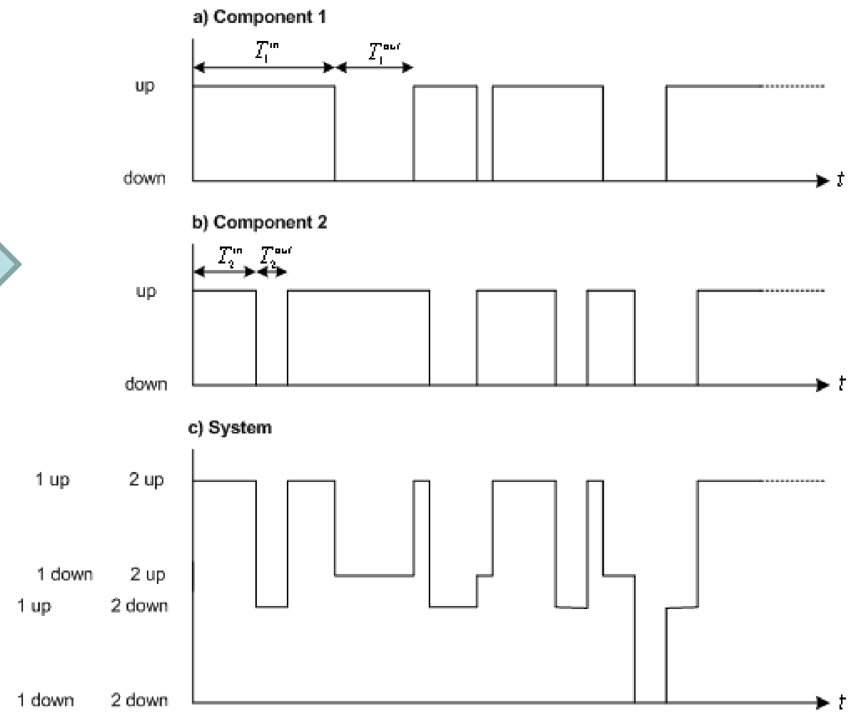
Component model



$$T_i^{in} = -\frac{1}{\lambda_i} \ln U_i^{in}$$

$$T_i^{out} = -\frac{1}{\mu_i} \ln U_i^{out}$$

Time sequence



Combining Monte Carlo Techniques with Object-oriented modeling

Advantages for reliability analysis:

- Monte Carlo simulation helps to overcome the problem of the **state space explosion**:

  Consider a system of $k=20$ components with two states (e.g. up state and down state). A "state enumeration approach", such as a "complete" fault tree, or a Markovian chain would have to consider $2^k = 2^{20} = 10^6$ system states!

- Object-oriented modeling helps to explicitly consider **time-dependent interactions** between the components and to integrate feedback loops, which is not possible in "static approaches" such as fault tree analysis.

Disadvantages for reliability analysis:

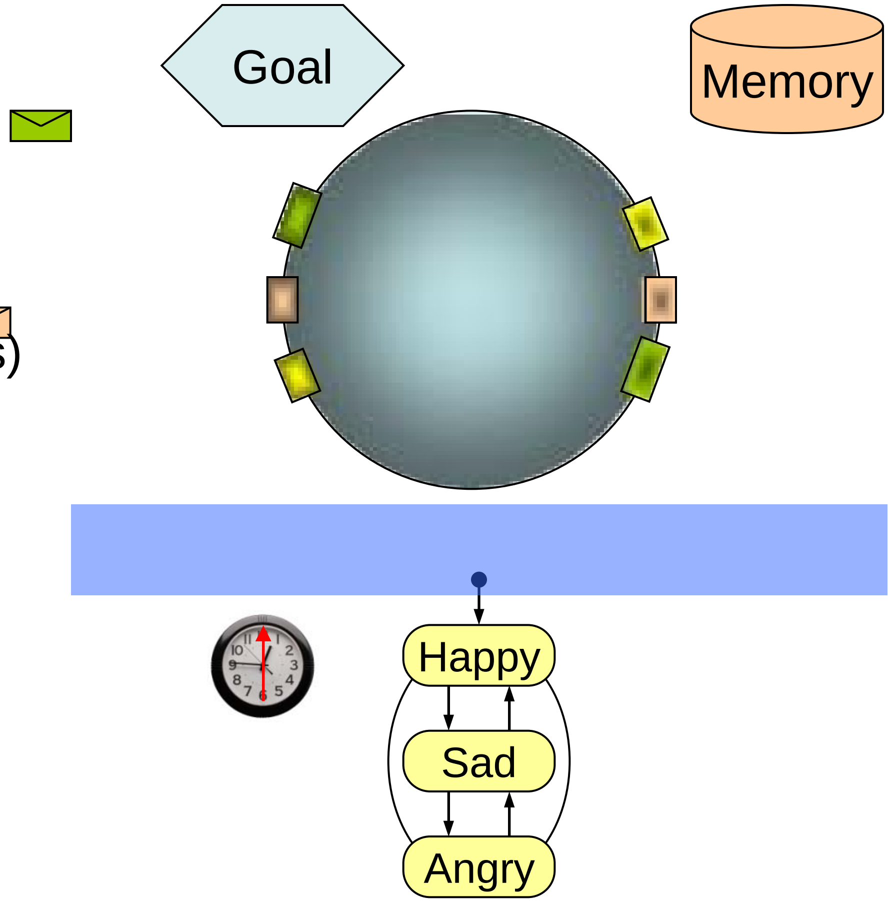
- The simulation primarily aims at calculating mean values. **Some critical scenarios might be missed.**
- Depending on the analyzed system, the **validation** of the simulated system behavior might be a **difficult task**, due to lack of operational experience regarding low-probability-high-impact scenarios.

Object-oriented modeling approach – framework

- Modeling the behaviour of the **components** (objects) and their interaction with the environment
- Stochastic simulation (Monte Carlo methods) of all components to investigate the **macro-behaviour** of the whole system
- In contrary to established methods for risk analysis (ETA, FTA) the observed scenarios and system states *s* are not predefined, but they emerge during the simulation (**emergence**)
- Frequency and consequence of events are determined "**experimentally**"

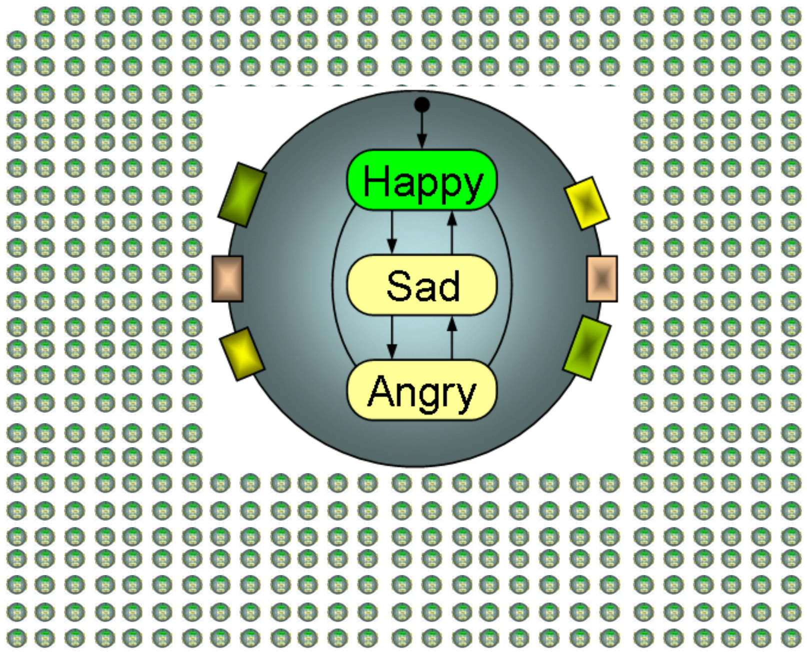
An object…

- **Has different states**
  (<u>F</u>inite <u>S</u>tate <u>M</u>achine, FSM)
- Is capable of interaction with its environment (e.g. other objects)
- has „receptors“ and „effectors“ for specific („messages“) and non-specific (environmental variables) signals
- Can act randomly
- May have a memory (learning)
- Can strive for a goal



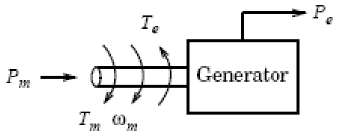
Simulation of N objects

- One single object does not tell us much about the behaviour of its macro-system
- Therefore every component of a system has to be modelled separately by an object
- By the computational simulation of all objects, the global system behaviour and the system states *s* emerge

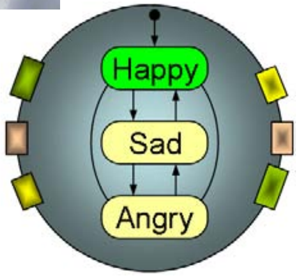


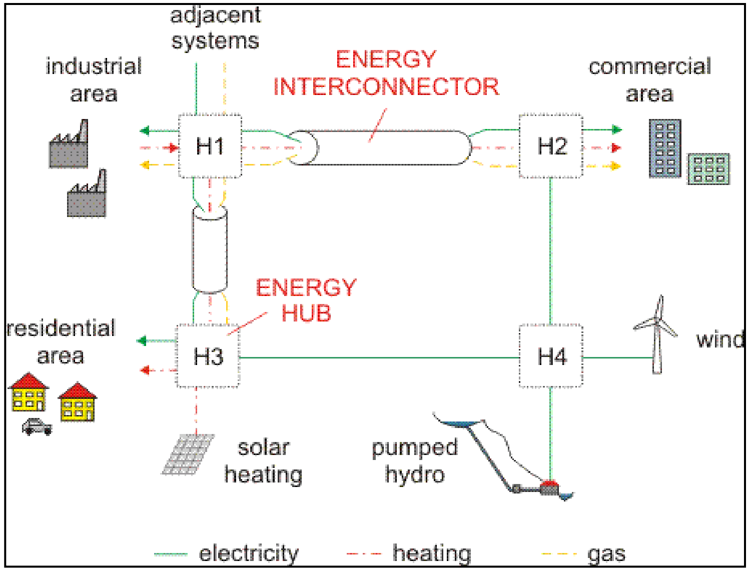
What can be represented by objects?

- Humans (e.g. operators)
- Components (e.g. turbine)
- Machines (e.g. power station)
- Whole systems
  (e.g. energy systems)

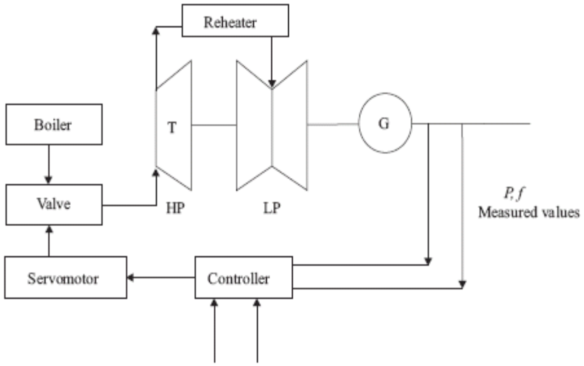


EEH, ETH Zurich





EEH, ETH Zurich

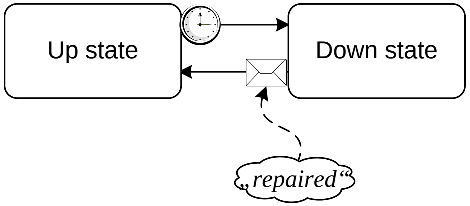


EEH, ETH Zurich

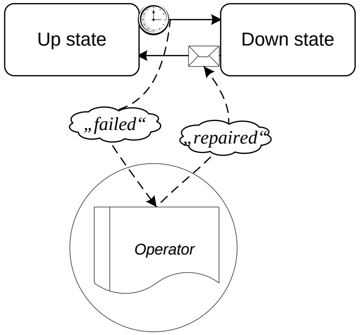
How to build a simplified object-oriented model for reliability analysis

1. Identify the components of the system
2. Determine the states of each component by making use of FSM, eg:

3. Determine the transitions between the states and their triggers (e.g. lapse of time or signal from outside)



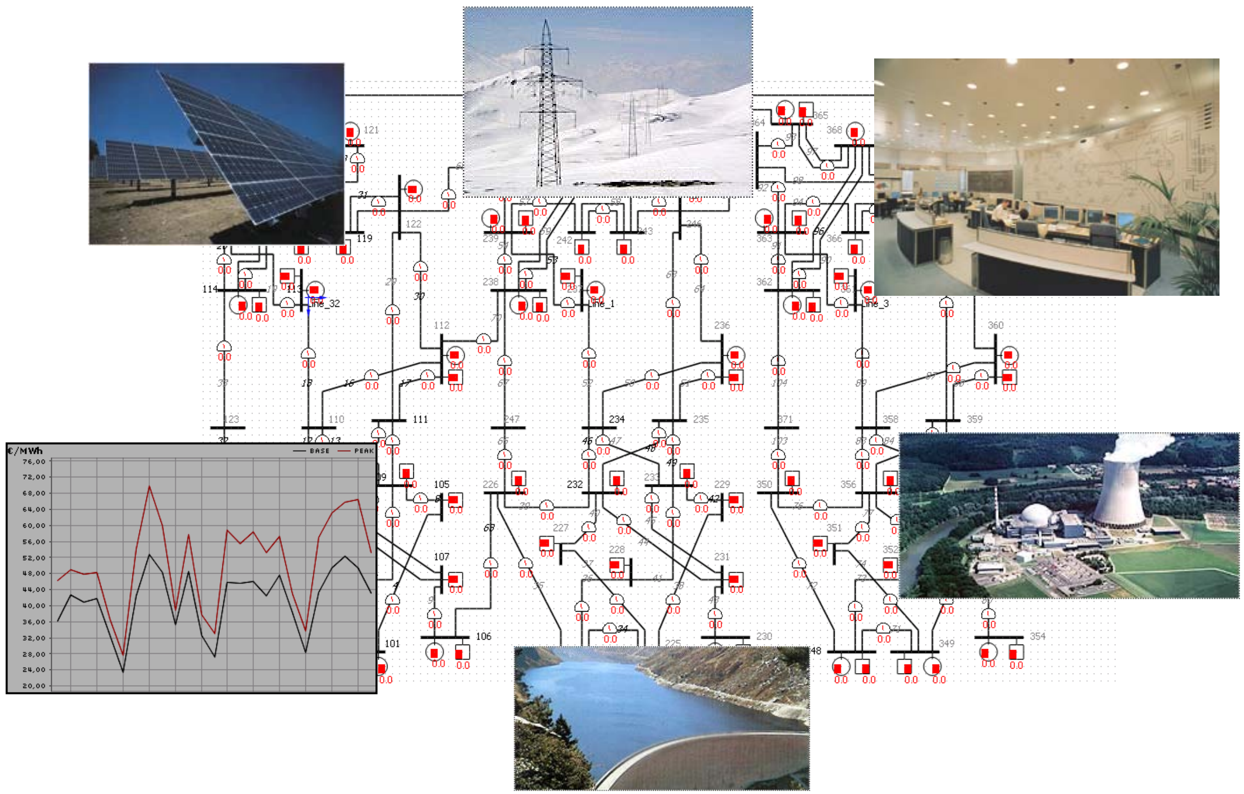
4. Establish the communication among the objects:



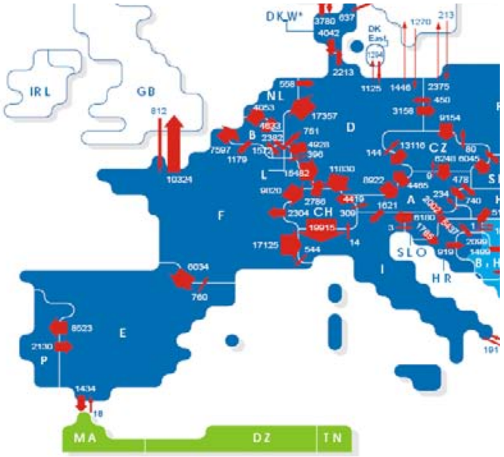
5. Simulate your model to generate the system states *s* and estimate $\overline{Q}$

Part II: Application to Complex Engineering Systems

Reliability Analysis of Electric Power Systems



Background and Motivation (I)



Source: UCTE 2006

Developments within the landscape of the European electric power system:

- *electricity market liberalization*
  - intricate, decentralized market structure with numerous competing players
  - extensive (international) trading without adequate technical upgrades
- *increasing amount of distributed generation and renewables*
  - less predictable power sources
  - variable load flow patterns
- *pervasive use of ICT*
  - more intelligent system operation
  - new risks due to interdependencies

⇨ **Unprecedented operational complexity**

Background and Motivation (II)

However:

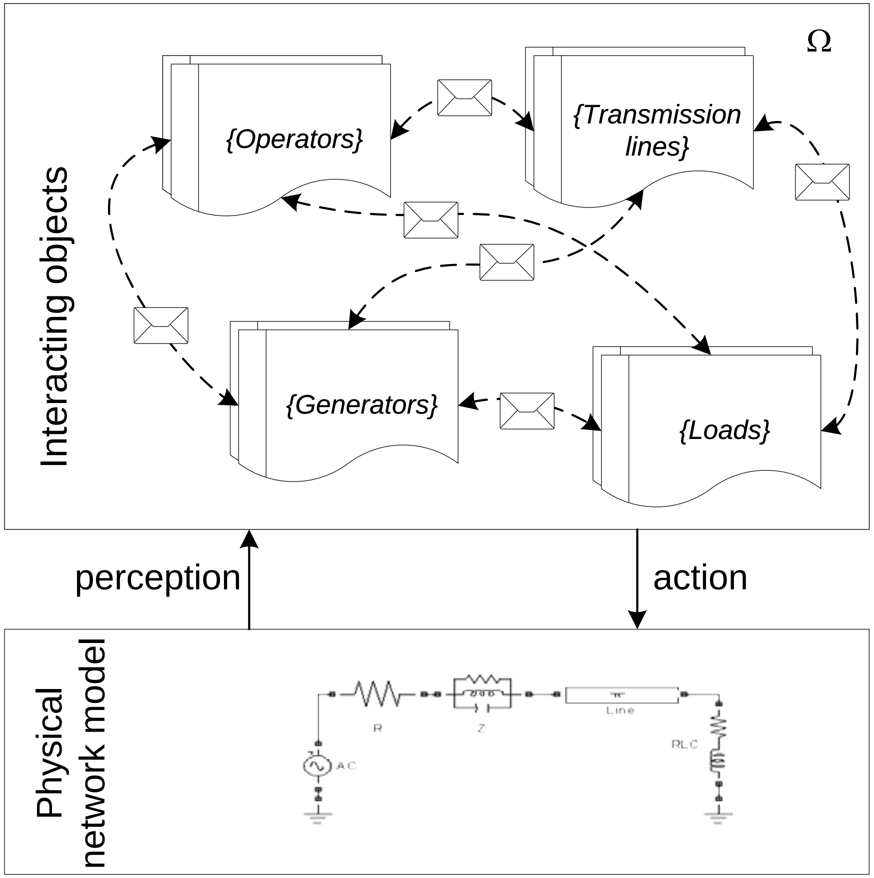
Reliability policies and assessment methodologies did not keep pace !

- *N-1 rule*
  - deterministic approach
  - predefined set of anticipated faults, focus on „worst-case“ contingencies
  - relevant failure states or the actual „worst-case scenario“ might be missed !!

- *Current probabilistic approaches*
  - limited to specific aspects (e.g. to a single component), or
  - oversimplified (e.g. cascading failures)

How would you build a simplified object-oriented model?

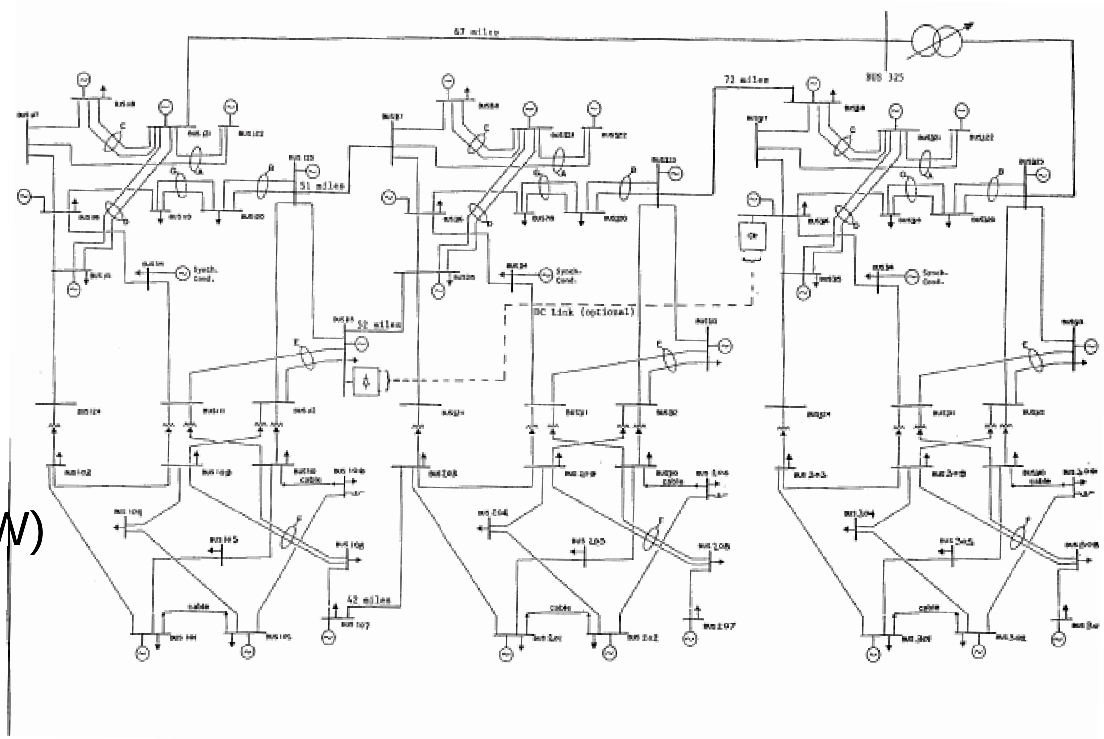
Modeling the Electric Power System – Two-layers approach:



The IEEE Reliability Test System 1996 (RTS `96)

➢ Basic system layout

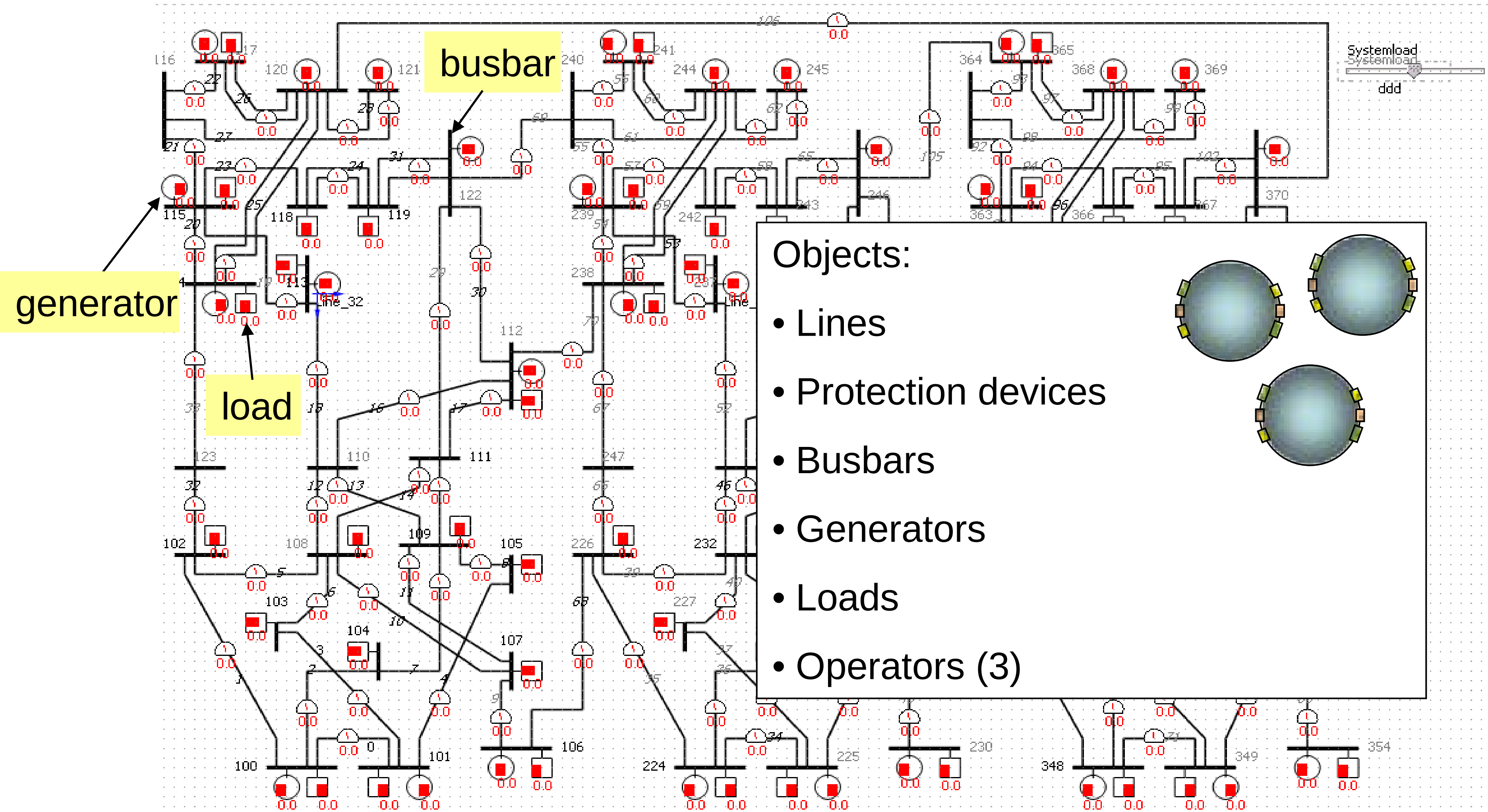
- 72 busbars
- 107 transmission lines
- 99 generators
- 51 loads
- Voltage levels: 230/138 kV
- Installed capacity:
  10‘215 MW (CH: ~12‘000 MW)
- Peak load:
  8‘550 MW (CH: 9‘650 MW)



➢ Available data

- physical component data: branch reactances, operational thresholds etc.
- load curves (hourly, daily, weekly)
- reliability data: component outage and repair rates, min. down times, etc.

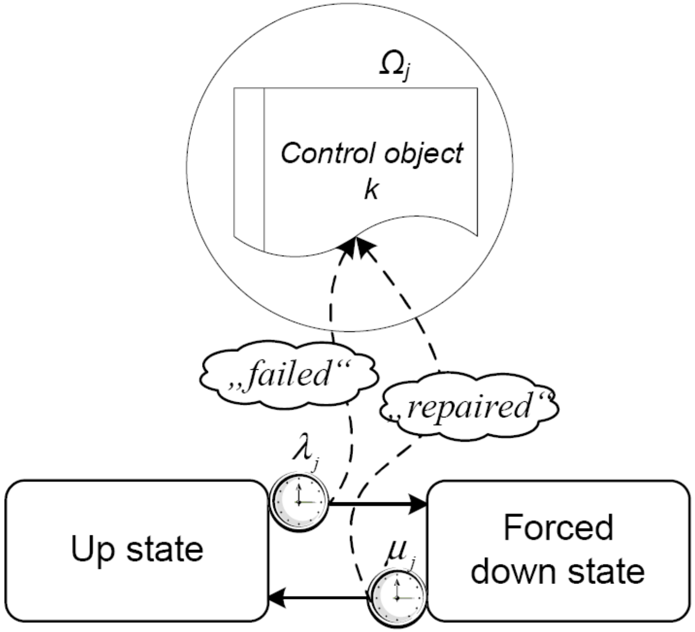
The RTS `96 - Implementation in AnyLogic



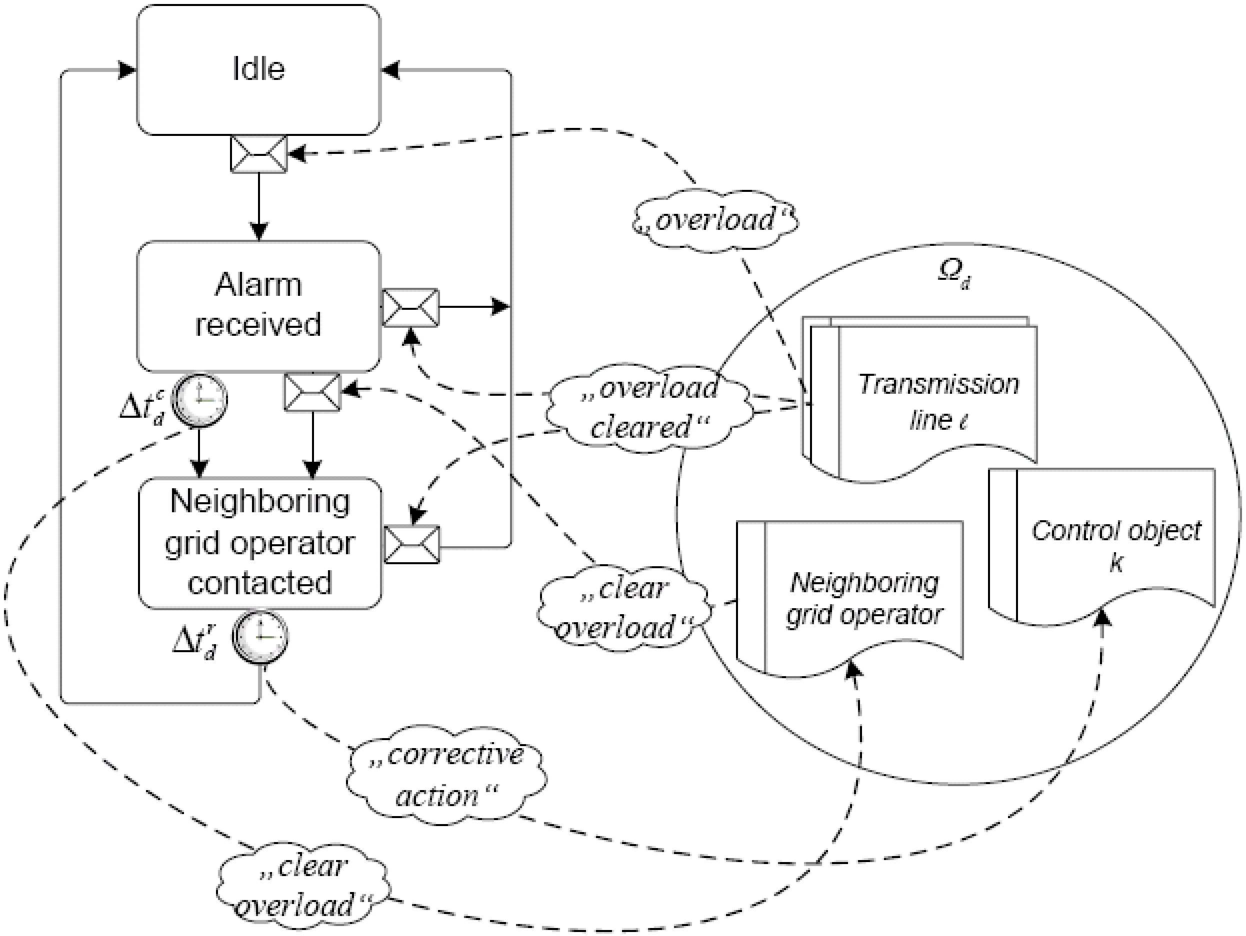
Objects:

- Lines
- Protection devices
- Busbars
- Generators
- Loads
- Operators (3)

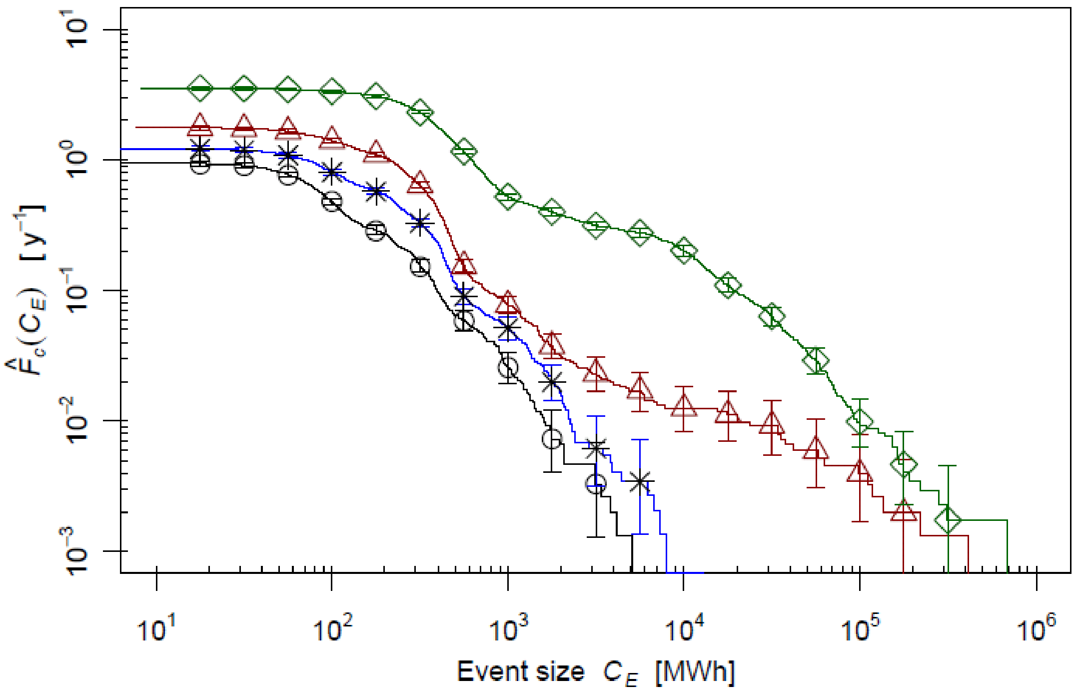
Objects: generators



Objects: sytem operators

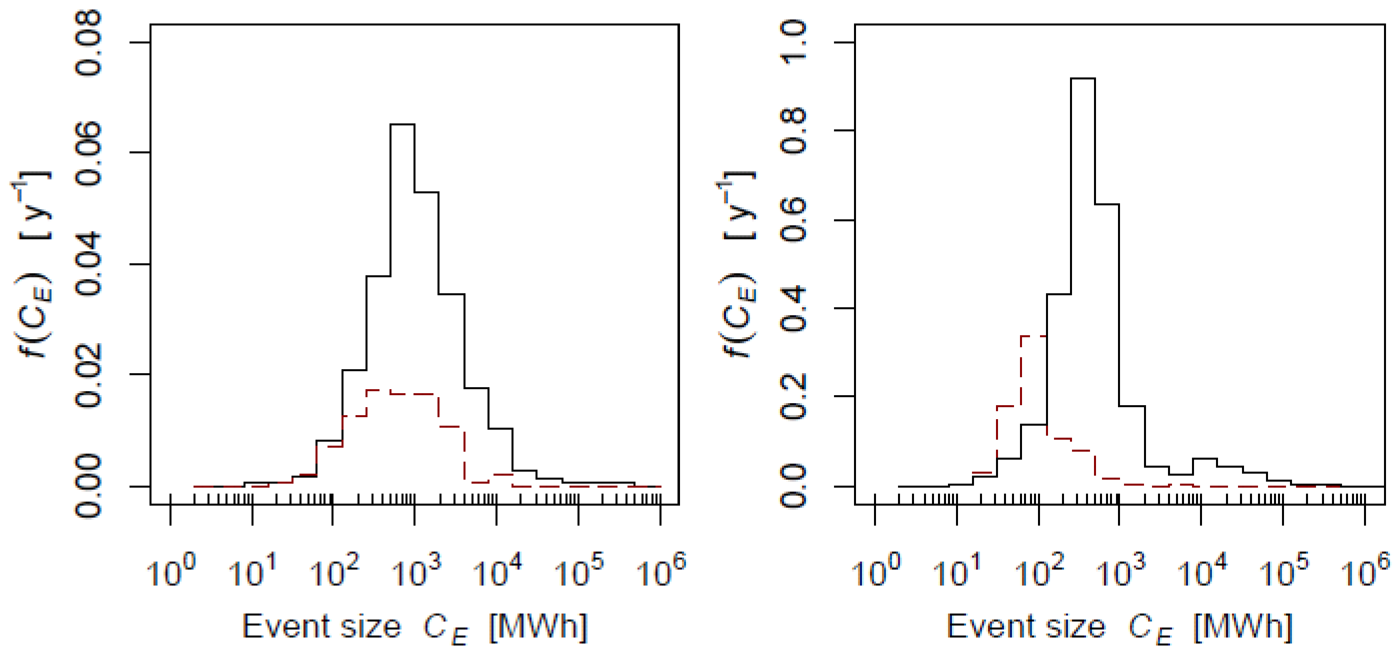


Results:
A) Expected Frequencies of Blackouts



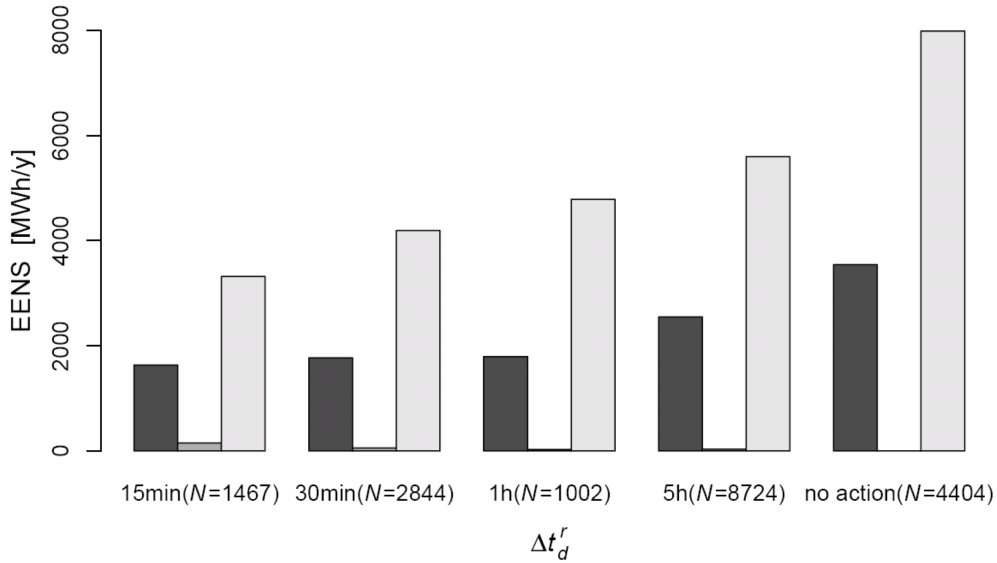
Complementary cumulative blackout frequencies for four different system loading levels L=1.0, 1.1, 1.2 and 1.37 (circles, stars, triangles and diamonds, respectively) without operator intervention. The error bars indicate the 90% confidence interval.

Results:
B) Blackout causes



Impact of increasing the system loading from L=1.0 (dashed line) to L=1.37 (continuous line) on the absolute frequencies of blackouts caused by generation inadequacy (left) and system splitting (right).

Results:
C) Influence of the operator response time on the system reliability



Influence of the operator response time on the EENS due to generation inadequacy (left, black bar), operator action (middle, dark-grey bar) and system splitting (right, light-grey bar) for L=1.37.

EENS: Expected Energy Not Supplied

Results:
D) Most disconnected lines in the Swiss system