# Discrete event simulation

Prof.dr.ir. Alexander Verbraeck
Professor, Faculty of TPM, TU Delft

# Overview

- What is discrete event simulation?
- Where does it fit historically?
- How does it differ from other types of simulation?
- What are the steps in a simulation study?
- What are the important aspects of simulation for infrastructure studies?
- Summary

# Simulation

Simulation is: [Shannon, 1975]

- a process of designing a model of a concrete system
- and conducting experiments with this model
- in order to understand the behavior of a concrete system
- and/or to evaluate various strategies for the operation of the system

# Simulation

Simulation is: [Shannon, 1975]

- a **process** of designing a model of a concrete system
- and conducting **experiments** with this model
- in order to **understand** the behavior of a concrete system
- and/or to **evaluate** various strategies for the operation of the system

What-if: parameters → output

# Why discrete simulation?

Instrument to:

- evaluate a systems design
- compare alternative solutions
- predict systems performance

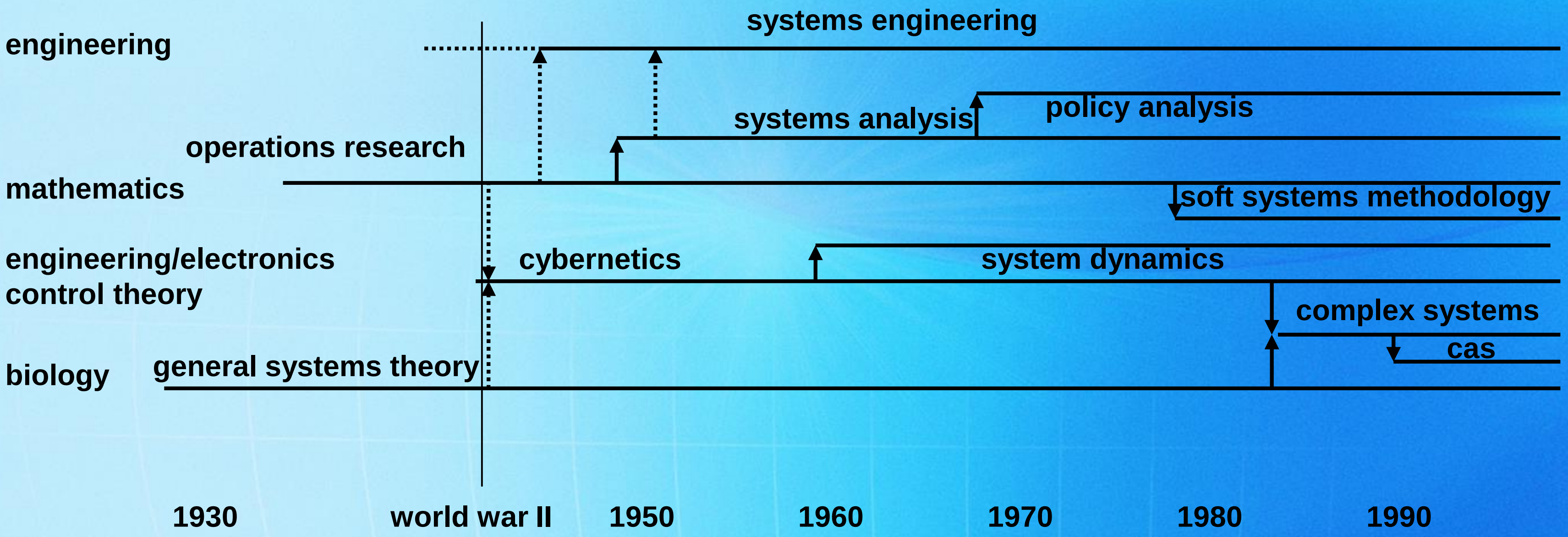
Mainly used for logistical problems:

- expected use of limited capacity or resources

In some cases more advanced use:

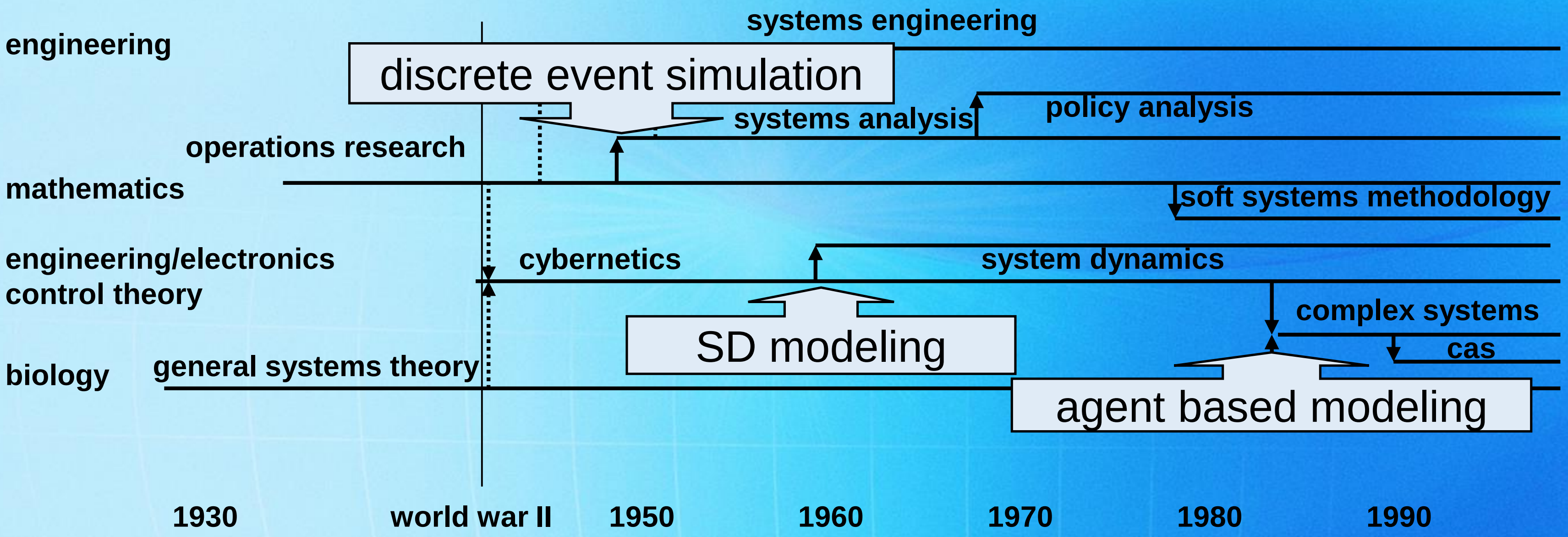
- sensitivity analysis
- optimization

# Systems thinking




Slide courtesy Els van Daalen, TU Delft

# Systems thinking




Slide courtesy Els van Daalen, TU Delft

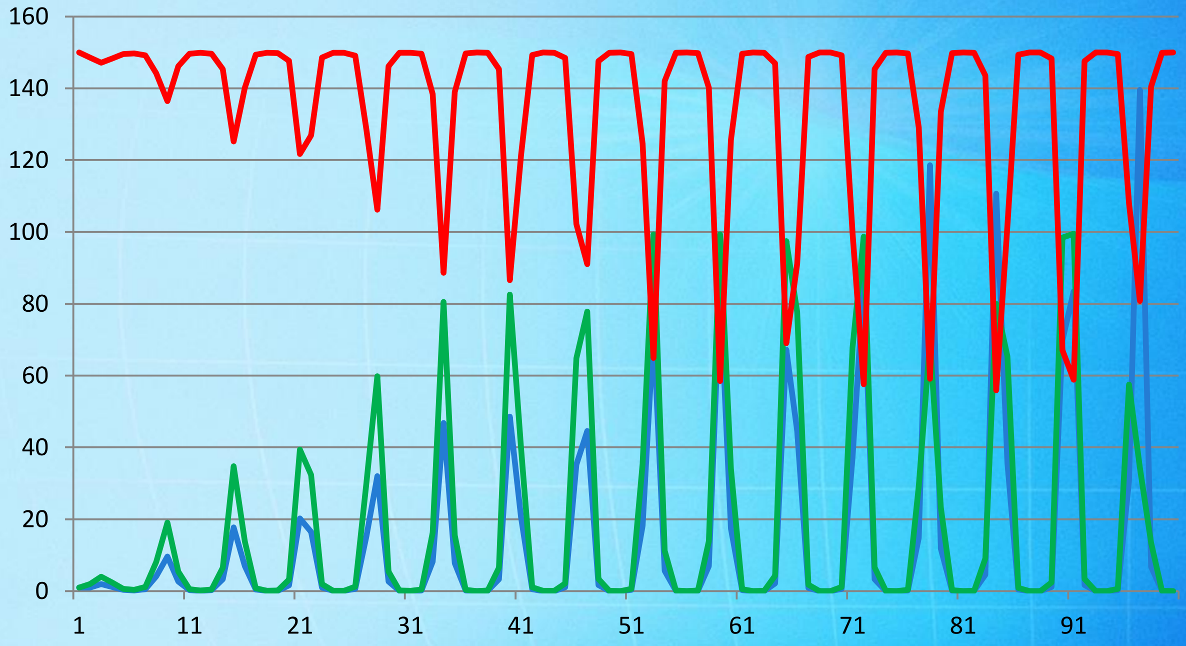
# Similarities and differences

- Models are used to study the **relationships** between variables
- Simulation models study the evolution of **variables** over time
- The values of the model variables at a given time is called the **state** of the model
- In **discrete-event** simulation models, state changes occur at an instant of time
- An **event** is a change in model state, occurring at an instant

Nance, 1981

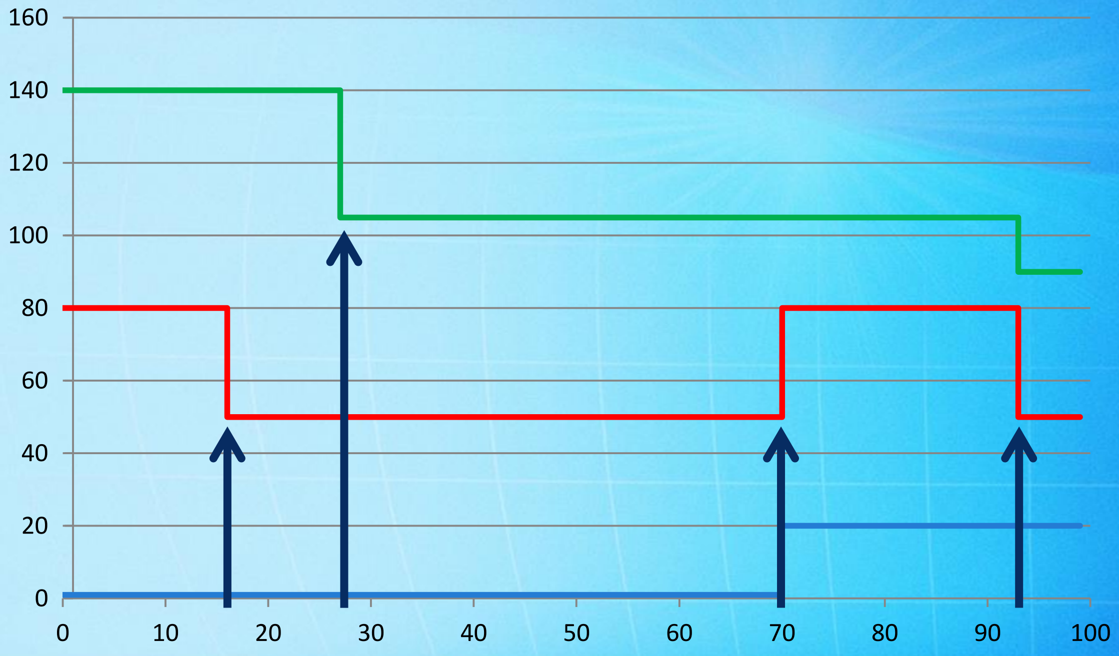
# Similarities and differences

In continuous models, state is a continuous function of time:



# Similarities and differences

In discrete-event models, state is a piecewise constant function over time:



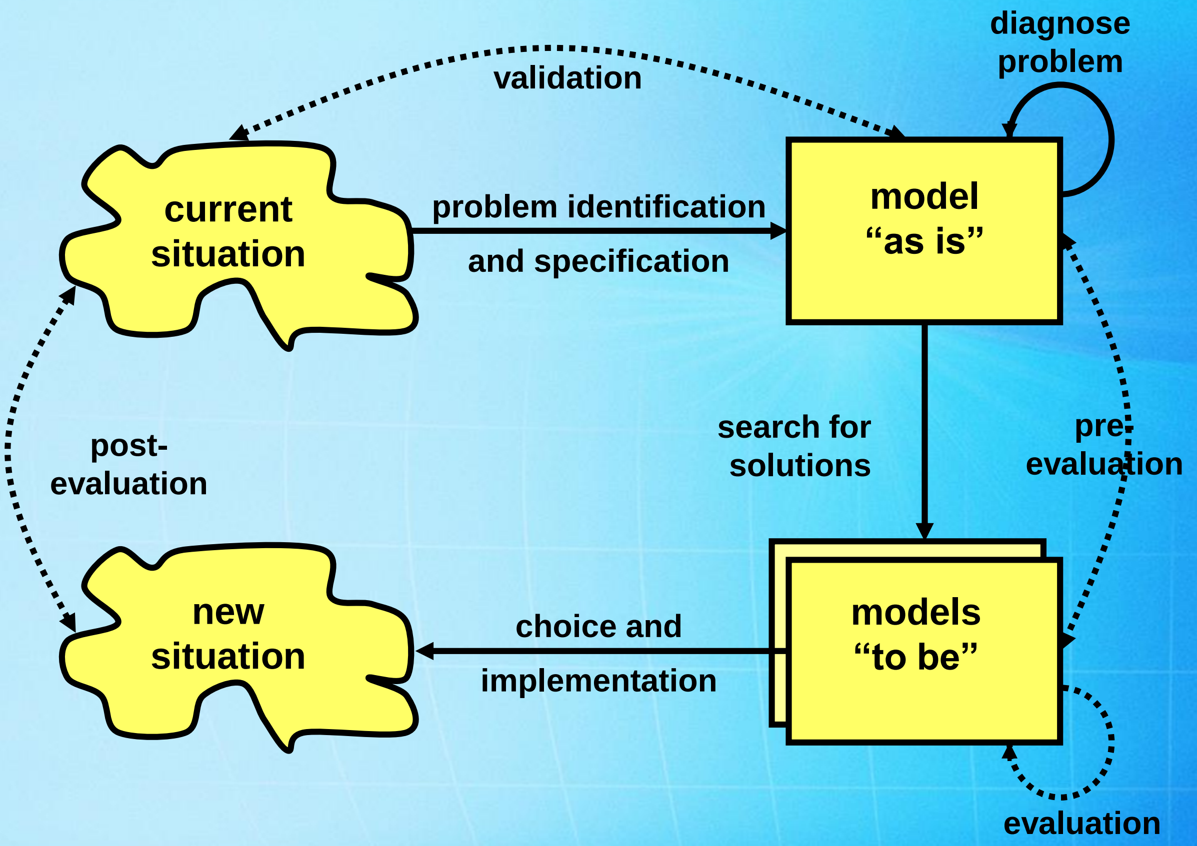
# Discrete changes over time

Very useful for:

- Queuing systems
- Resource usage
- Transportation
- Logistics and warehousing
- Control systems
- etc.

For all these systems it means that we have to focus on the **events**, i.e. the start and the end of processes rather than the evolution of the process itself

# Simulation model lifecycle



# Steps in a simulation study

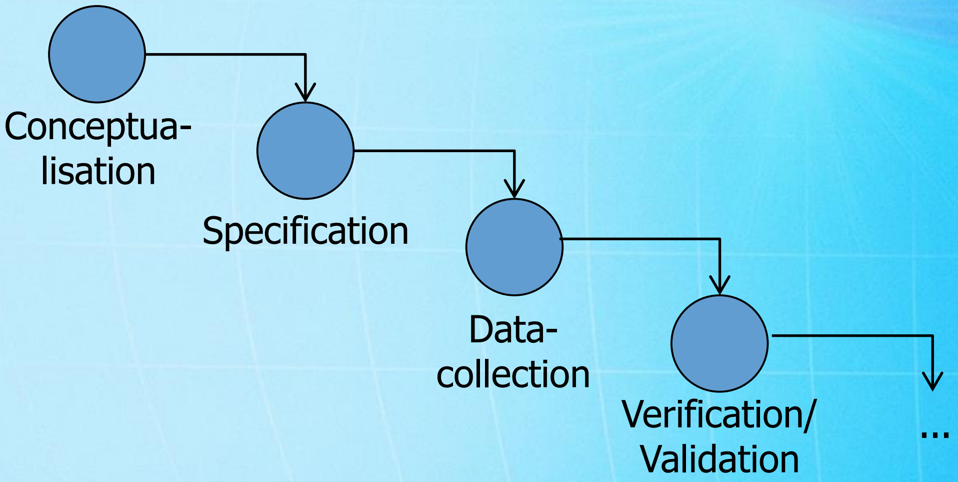
- Conceptualization
  - Demarcation
- Specification
  - Reduction
  - Data gathering
  - Model building
- Verification and validation
- Experimentation
- Analysis
- Alternative generation
- Model adaptation
- Conclusions and reporting

# Simulation project plan

Traditional: waterfall model

or iterative modeling

But better: incremental modeling

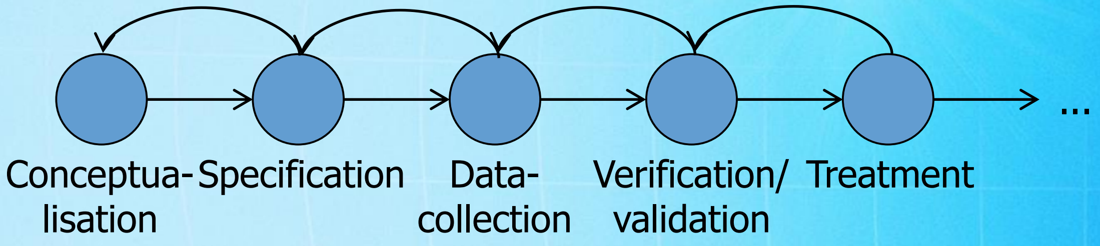


# Simulation project plan

Traditional: waterfall model

or iterative modeling

But better: incremental modeling

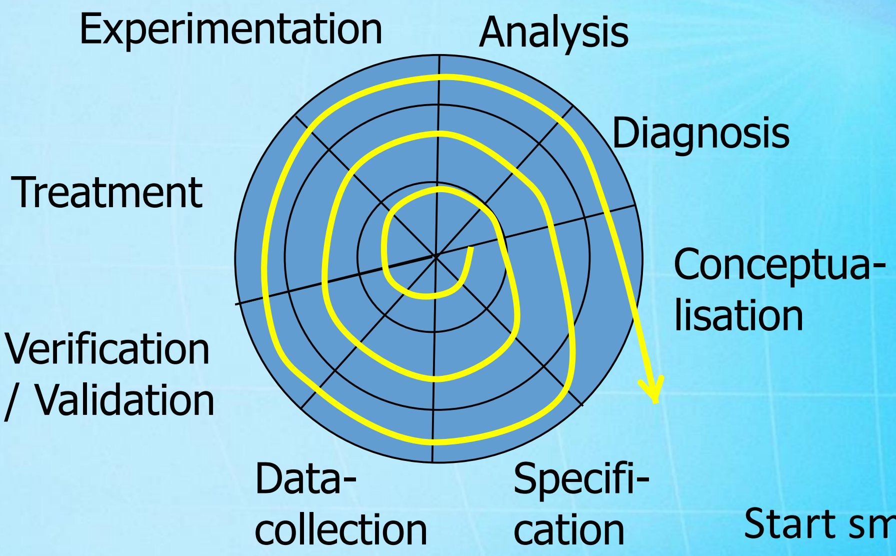


# Simulation project plan

Traditional: waterfall model

or iterative modeling

But better: incremental modeling




Start small...

# Conceptualisation

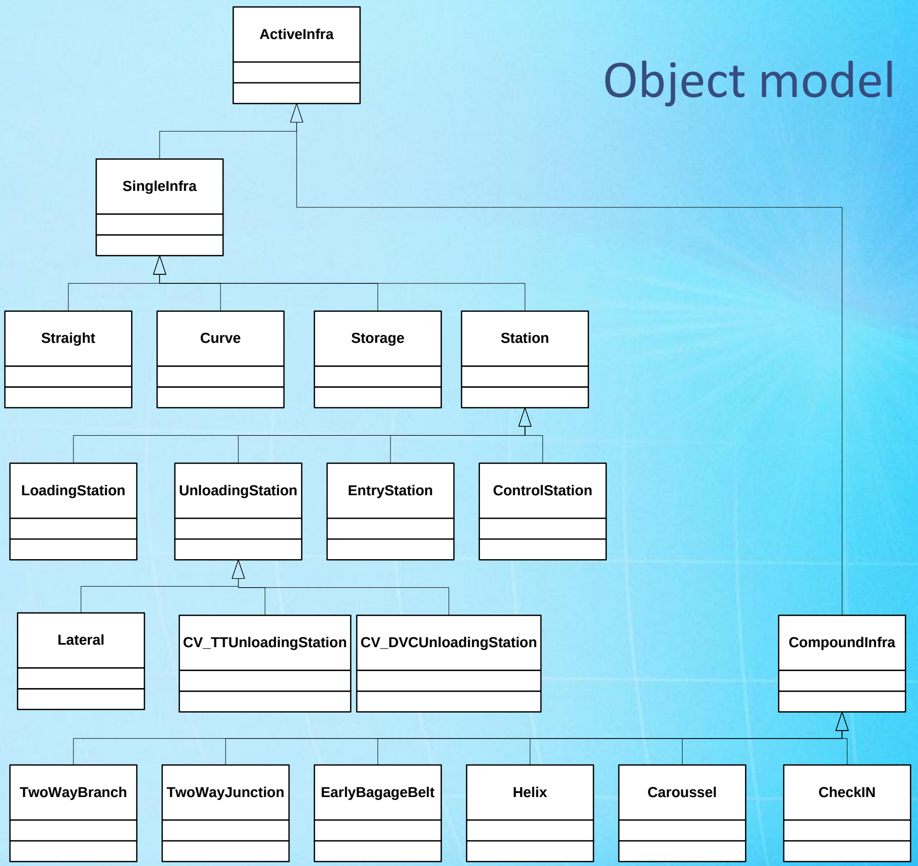
**Output**:

a number of conceptual models that can be used to describe the system

- Demarcation of the system
- Language by which the system can be described:
    - object based (object model)
    - process based (process model)
    - time based (event list)

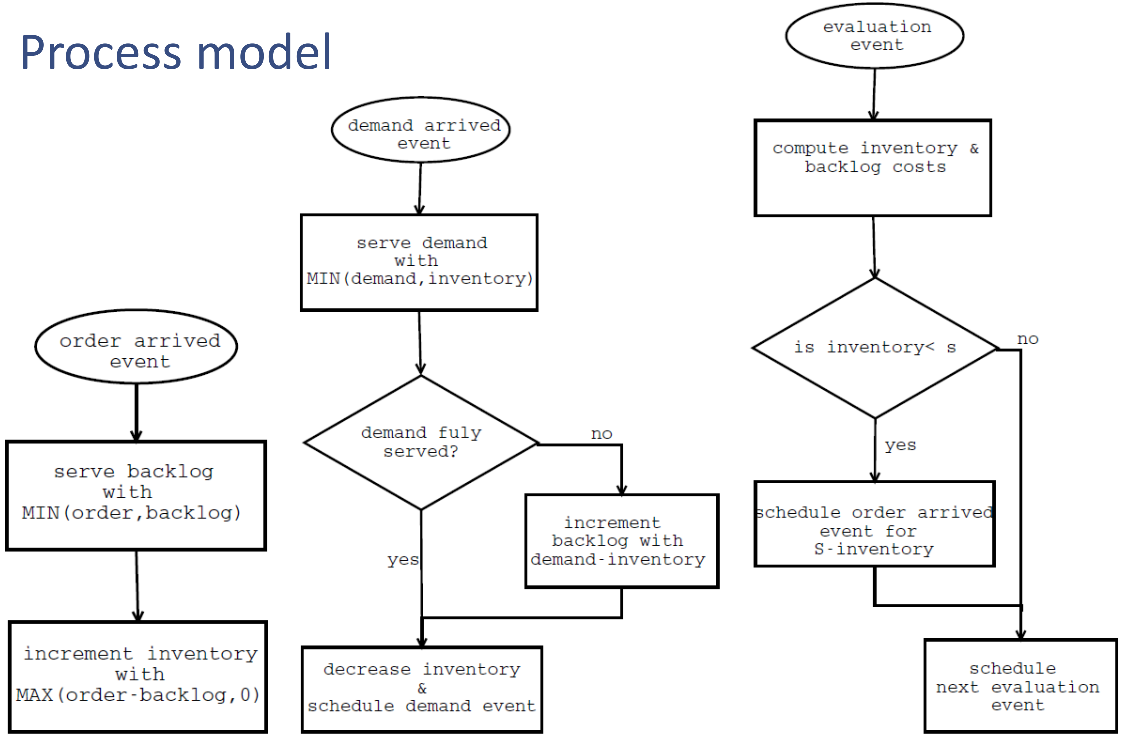
# Conceptualisation

## Object model



# Conceptualisation

## Process model




(a) order event (b) demand event (c) evaluation event

# Specification

**Output**:

working model that can be experimented with

- Reduction of the model
- Specification of model
- Detailed input/output specification
- Data gathering
- Build simulation model

# Data for discrete simulation

- Data for:
  - generators of items
  - process durations in the model
  - resource availability

- How to gather data:
  - historical sources
  - expert opinions
  - measurements
  - analogous systems

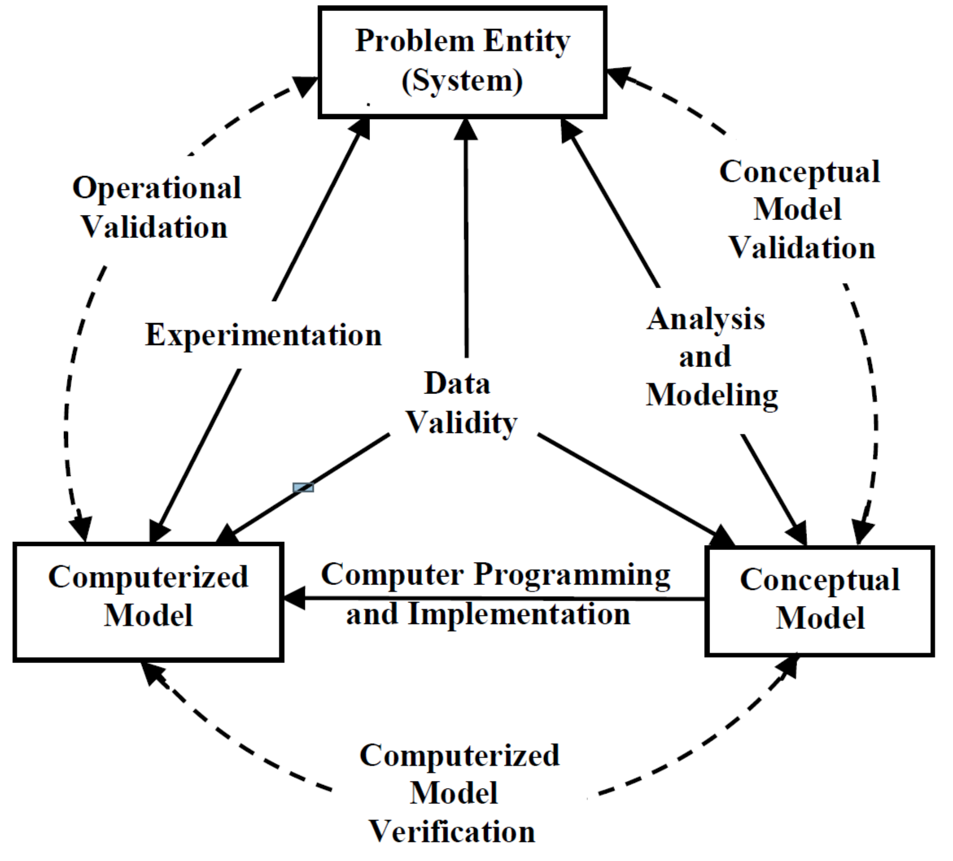
# Verification/validation

**Output**:

simulation model that is correct and is a good representation of the real system

- Verification (correct representation of conceptual model)
- Validation (models represents reality):
  - *structural*: testing of hypotheses on the model
  - *operational*: compare values to real system values
  - *expert*: analysis of the model by experts

# Verification/validation




Sargent, R.G. (2009). VERIFICATION AND VALIDATION OF SIMULATION MODELS. In: M. D. Rossetti, et al. (Eds.) *Proceedings of the 2009 Winter Simulation Conference*, IEEE, 2009, pp. 162 - 176.

# Experiment specification

**Output**:

the run control conditions under which the system, or the model of it, is experimented with or observed

- Number of runs
- Run length
- Start-up time
- Values of input parameters
- Output parameters to be calculated

# Analysis and diagnosis

**Output**:

results of analysis and diagnosis of the experiments with the model of the current situation

- Comparing alternatives
- Statistical analysis
- Current bottlenecks (long queues, idle resources, etc.)
- Sensitivity analysis for stability of results

# Analysis and diagnosis

## Statistical analysis

**T-Test**

[DataSet0]

**Group Statistics**

| | ScenarioID | N | Mean | Std. Deviation | Std. Error Mean |
|---|---|---|---|---|---|
| GemAantalnWachtrijAan meldpunt | 1.00 | 30 | 1.9531 | .95607 | .17455 |
| | 2.00 | 30 | .0983 | .02085 | .00381 |

**Independent Samples Test**

| | | Levene's Test for Equality of Variances | | t-test for Equality of Means | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | 95% Confidence Interval of the Difference | |
| | | F | Sig. | t | df | Sig. (2-tailed) | Mean Difference | Std. Error Difference | Lower | Upper |
| GemAantalnWachtrijAan meldpunt | Equal variances assumed | 15.133 | .000 | 10.623 | 58 | .000 | 1.85479 | .17460 | 1.50530 | 2.20428 |
| | Equal variances not assumed | | | 10.623 | 29.028 | .000 | 1.85479 | .17460 | 1.49772 | 2.21186 |

# Infrastructure simulation

- Replicate system components 1:1 as simulation model components
- Use of hierarchy to build a model "bottom-up"
- Libraries of components available in multiple simulation languages
- Infrastructure capacity and usage ↔ resource capacity and usage
- Animation can help in building, debugging and presenting
- All simulation libraries have components that gather many different statistics

# Conclusions

- Discrete-event simulation:
  - state change over time
  - events
  - piecewise constant state
  - fast execution
- Model cycle:
  - incremental building
  - building blocks
  - hierarchy, flow, process
- Data-intensive
  - stochastic
  - statistics for input and output

# Thank you for your attention!

Please post any questions you may have
on our discussion forum