Discrete Event Simulation

IN2045

Chapter 0 - Simulation

Dr. Alexander Klein
Stephan Günther
Prof. Dr.-Ing. Georg Carle

Chair for Network Architectures and Services
Department of Computer Science
Technische Universität München
http://www.net.in.tum.de

Topics

- Introduction of basic terms
  - Model, systems, simulation…
- Evaluation spectrum
- When to use simulation
- Typical use cases for simulations

Modeling & Simulation

Model: A representation of a system (or: entity, process, …)

Simulation: The process of exercising a model to characterize the behaviour of the modelled system / entity / process over time

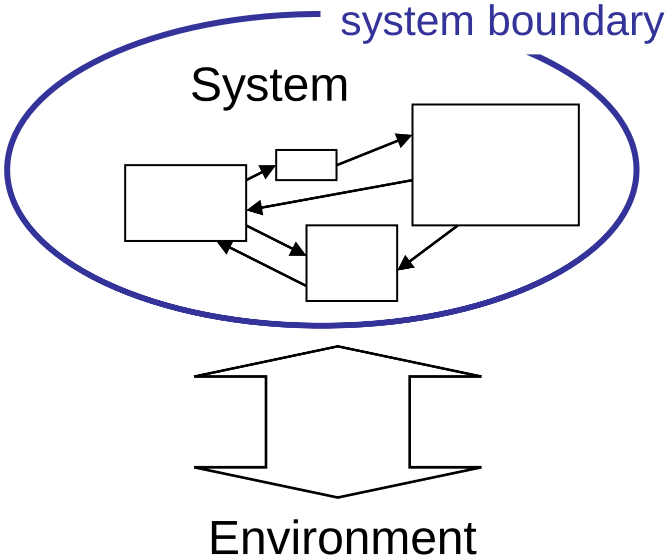
Computer simulation: A simulation where the system doing the emulation is a computer program

What is a system?

System:

- Actually, a very vague notion—pretty much anything can be a system!
  - ‘A system is what is distinguished as a system.’ (Brian Gaines)
- A system is something that we want to see separated from its environment through an (arbitrarily chosen) **boundary:**
  - **Inside** the system:
    - [opaque, i.e., black box — less interesting]
    - or some structure, mechanisms, rules
    - or even sub-systems

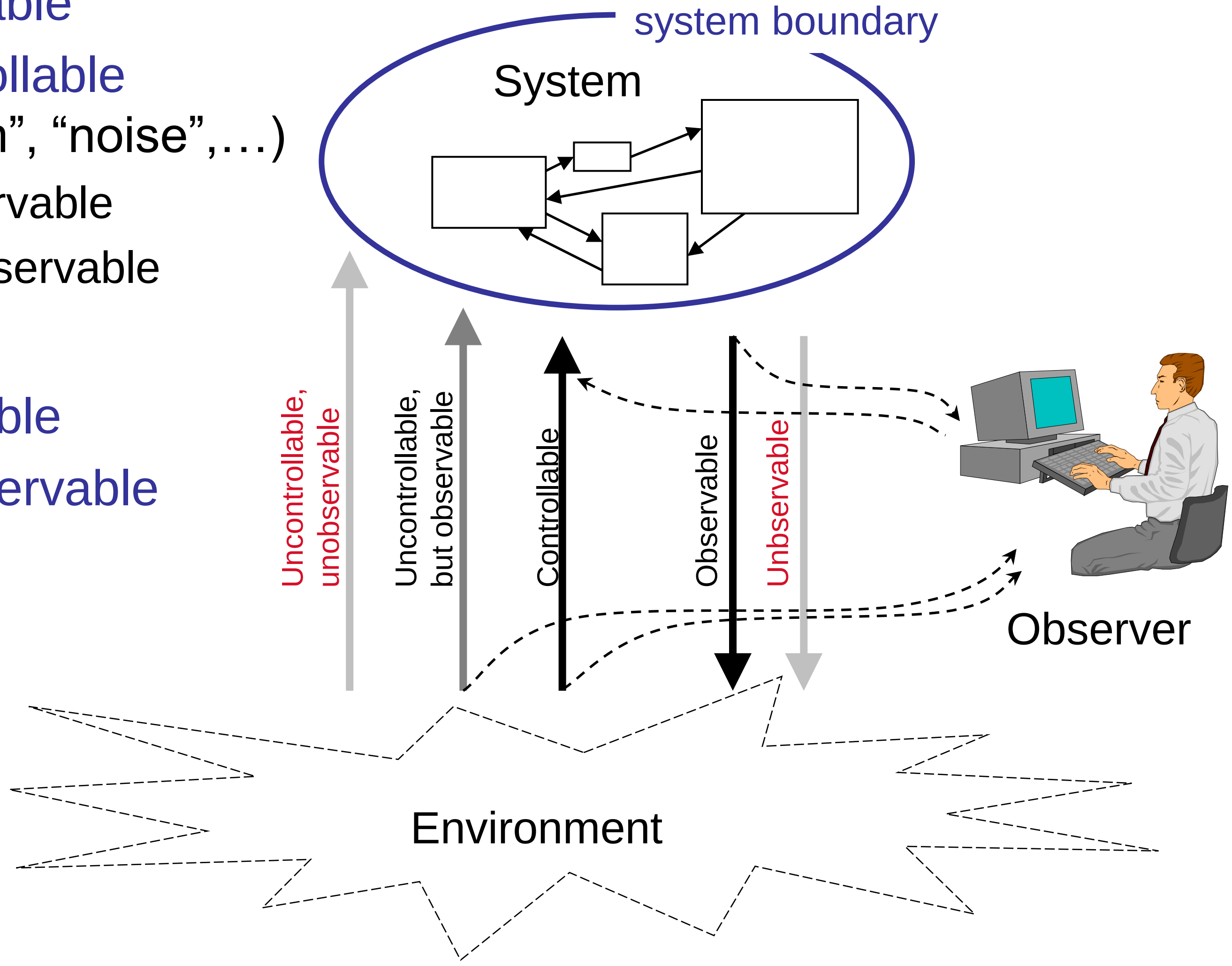
    (usual case)
  - **Outside** world (not part of the system!):
    - Environment, context
    - Interaction: Input from outside world, output into outside world



- A system has a **purpose**
  - Nobody defines something as a system without some purpose in mind

System, environment, observer

- Input
  - Controllable
  - Uncontrollable ("random", "noise",…)
    - Observable
    - Unobservable
- Output
  - Observable
  - Non-observable

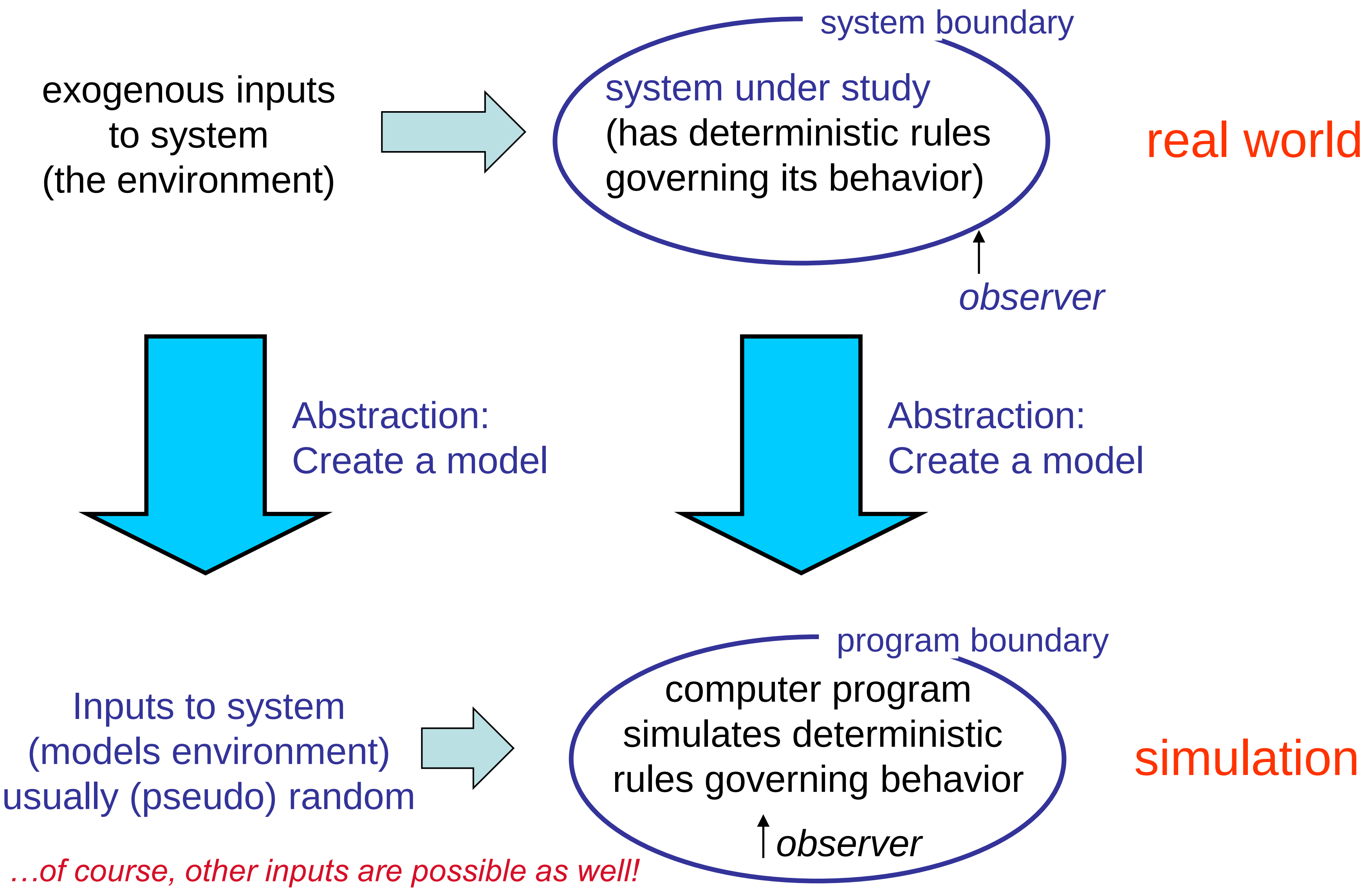


What is a model?

A **model …**

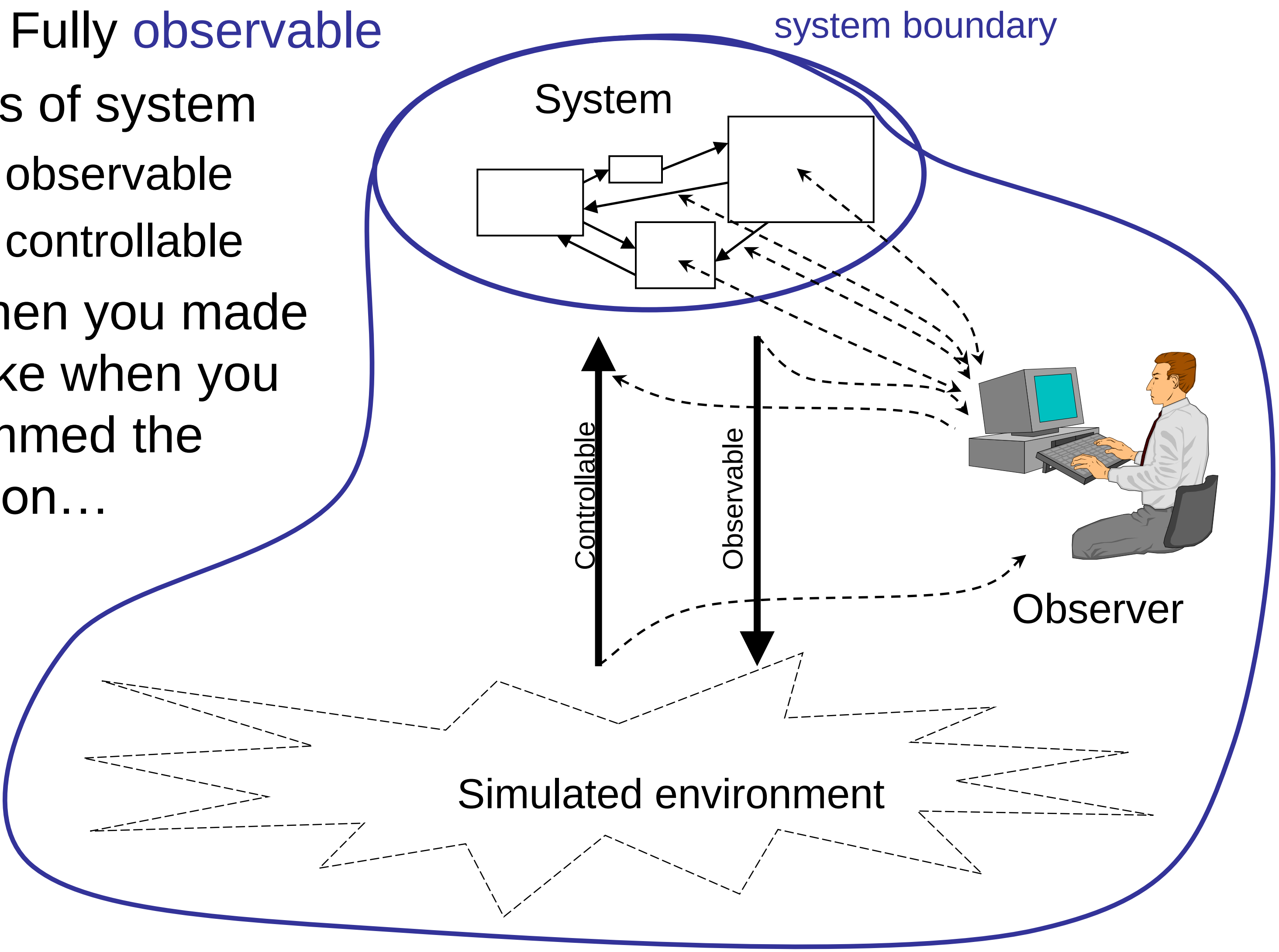
- is a system, too!
- mimics behaviour/characteristics of another system
- is material or immaterial
  - Material model: architecture models, or e.g.,:
  - Models we'll be talking about: normally immaterial
- allows experimental manipulation
- Purpose:
  - Simplification of original model: Reduction of complexity
  - Retaining those characteristics of original model that are important to the observer

Simulation is exercising a model (actually: two!)



System, environment, observer in simulation

- Input: Fully controllable
- Output: Fully observable
- Internals of system
  - Fully observable
  - Fully controllable
- If not, then you made a mistake when you programmed the simulation…



Ways to study a system

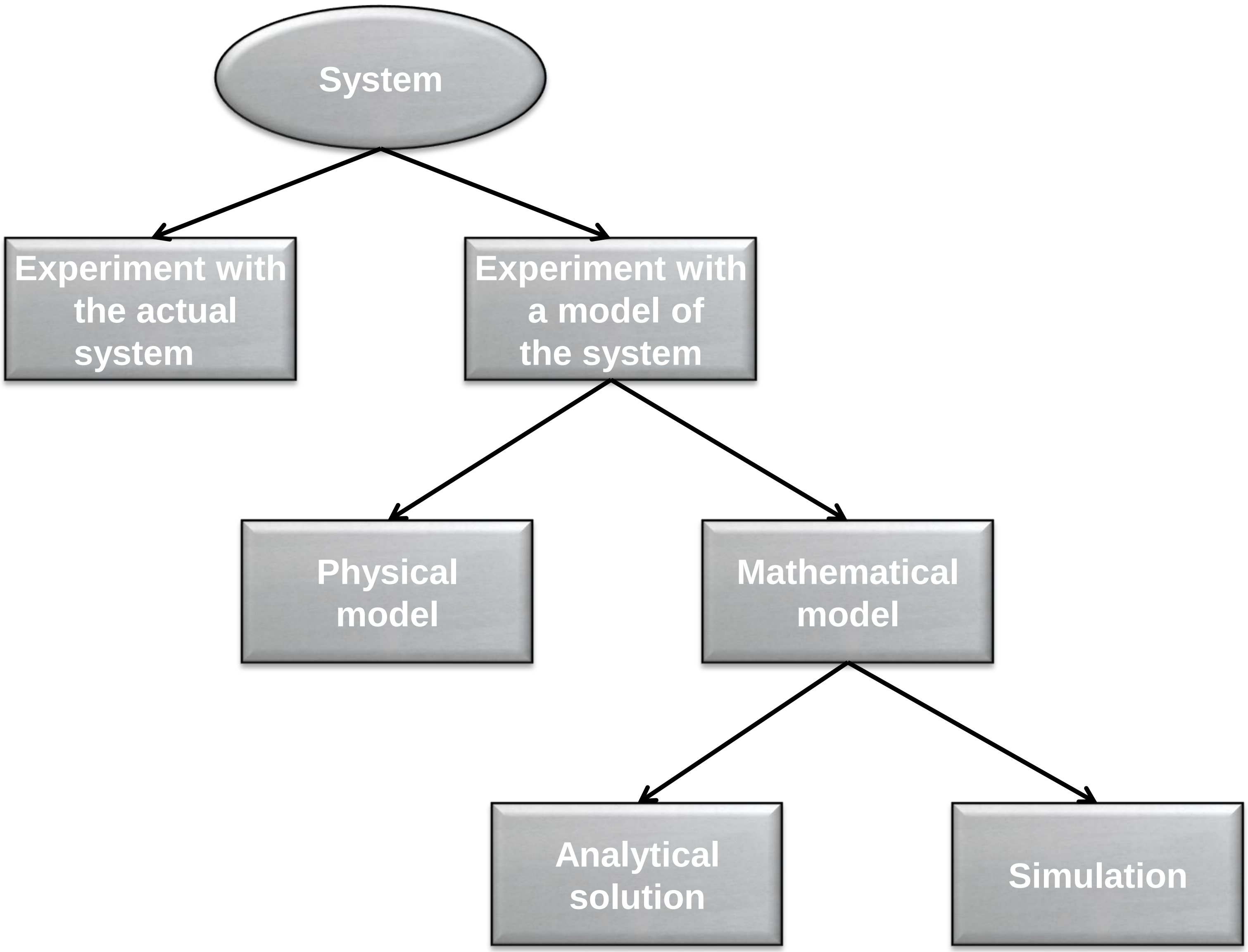


Figure adopted from LK 1.1

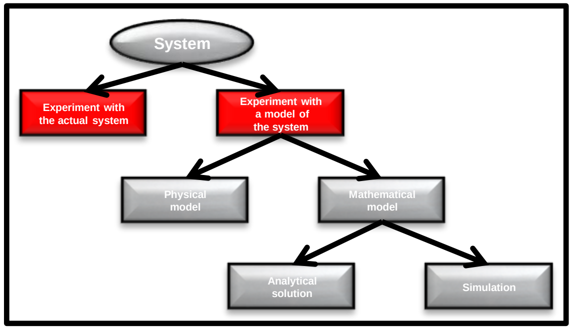
Ways to study a system (1/3)

Experiment with the actual system:

- Advantage:
  - Study is always valid
- Disadvantage:
  - Often too costly
  - Disruptive to the system
  - System might not even exist
  - Long-term study not feasible

Experiment with a model:

- Advantage:
  - Does not disrupt the actual system
  - No risks of system damage
- Disadvantage:
  - Accurate reflection of the actual system?
  - Is the model valid?



Ways to study a system (2/3)

Physical model:

- Advantage:
  - Often very accurate
- Disadvantage:
  - Usually expensive
  - Cannot be applied to all systems
  - Typically used for engineering or management systems
  - Smaller scales may result in different behavior

Mathematical model:

- Advantage:
  - Simple to apply
  - Allows abstraction of complex systems by using logical and quantitative relationships
  - Can be used for verification
- Disadvantage:
  - Accurate reflection of the actual system?
  - Is the model valid?
  - Are all relevant characteristics considered?

Ways to study a system (3/3)

Analytical solution:

- Advantage:
  - Often faster than simulation
  - Optimal for non-complex systems
  - Can be used for verification
- Disadvantage:
  - Complex systems are hard to describe by a mathematical model
  - Analytical solution usually have to apply higher levels of abstraction

Simulation:

- Advantage:
  - Simple to apply
  - Very flexible in terms of complexity
- Disadvantage:
  - Accurate reflection of the actual system?
  - Are all relevant characteristics considered?

Alternatives to simulation (1/2)

Evaluation spectrum:

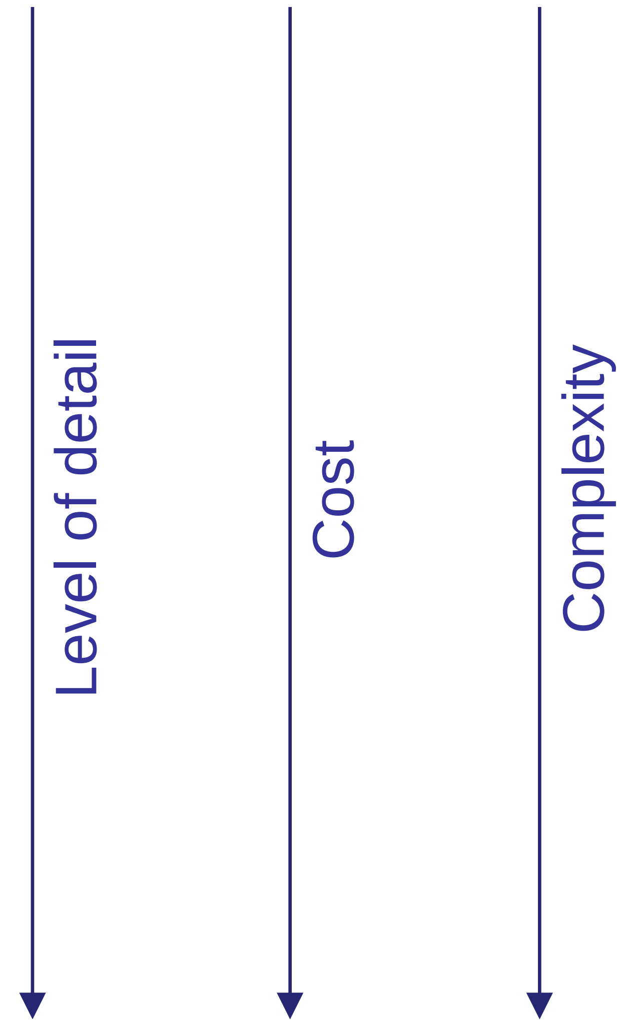
- Purely mathematical model
  using closed-form expressions
- Numerical models

  ↑ less detailed

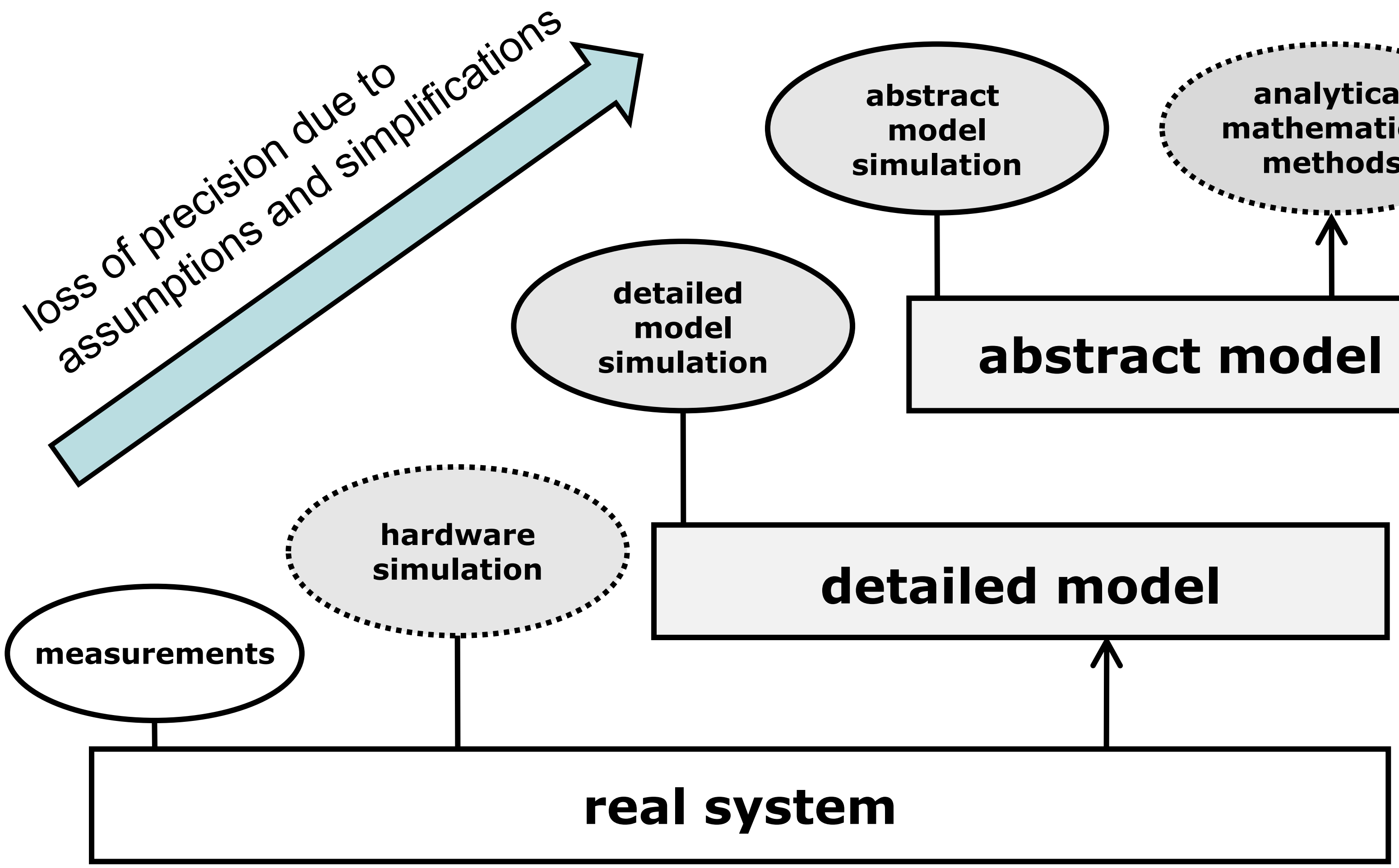
- **Simulation**

  ↓ more detailed

- Emulation
- Prototype
- Operational system



Evaluation spectrum:



When to use simulations (1/2)

- It may be too difficult, hazardous, or expensive to observe a real, operational system.
  Examples:
  - Virus epidemy
  - New routing protocol in the Internet
- There is only one real system, but we want to quickly evaluate alternatives and what-if scenarios.
  Examples:
  - Different router configurations
  - Different types of network traffic (realistic, low rate, full rate,…)

When to use simulations (2/2)

- Parts of the system may be unavailable / not be observable. Examples:
  - Internals of a biological system
  - Internals of a switch chip
- The original system runs on a very slow timescale, and/or we want to make predictions.

  Examples:
  - Climate predictions (10s to 1000s of years)
  - Milky way eating Sagittarius dwarf (100 mio years and more)
- It may be too difficult or intractable to model a system in detail using only closed-form expressions ("formulae").

  Examples:
  - Physical processes in atmosphere (weather, climate,…)
  - n-bodies problem, $n \geq 3$
  - Complex network with many TCP hosts

Simulation: Advantages

- Save lives
- Save money
- Save time (?)
  - Buying hardware, connecting and configuring a huge test network takes longer than setting up a simulation (…usually)
- Development / Find bugs (in design) in advance
  - The earlier a bug is detected, the less its removal will cost
- More generally applicable than analytic/numerical techniques
- Detail: can simulate system details at arbitrary level

Simulation: Drawbacks

- Caution: Does model reflect reality? Or is it too oversimplified?
- Large scale systems = Lots of resources to simulate, especially if accurate simulation is required
- Large scale systems = Lots of resources for simulator:
  - May be slow (computationally expensive: 1 min real time could be hours of simulated time!)
  - May eat huge amounts of RAM
  - May write out gigabytes of output (…which needs to be analyzed after!)
- It's an art: determining right level of model complexity
- Statistical uncertainty in results:
  - Was the simulation accurate/detailed enough?
  - Are the observed effects just artefacts/statistical outliers? Remember: Some input comes from a (pseudo-)random generator!

Use cases and applications for simulations

- Analyze systems before they are built
  - Reduce number of design mistakes
  - Optimize design
- Analyze operational systems
  - What-if scenarios
  - Find reasons for aberrant behaviour
- Create virtual environments for training, entertainment
  - Flight simulators, battlefield simulators
  - …in fact, almost all computer games are simulations!

Applications (1): System Analysis (focus of lecture!)

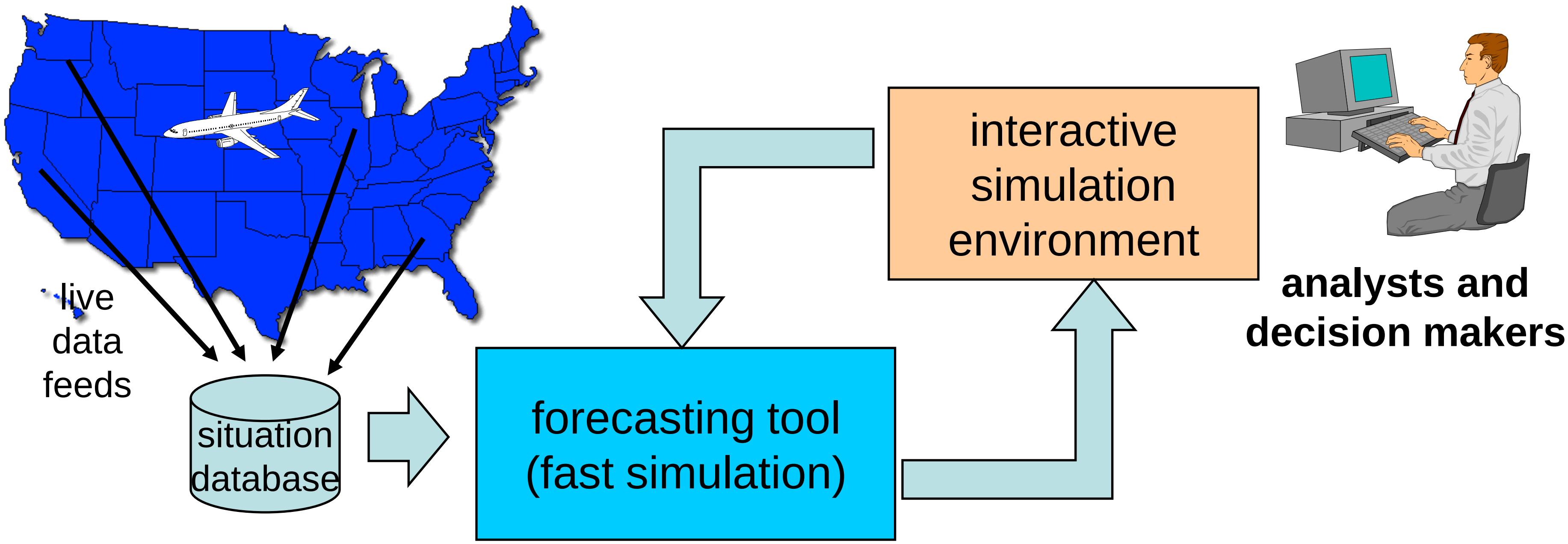
"Classical" application of simulation; here, focus is on "discrete event" simulation

- **Telecommunication networks** (focus of lecture!)
- Transportation systems
- Electronic systems (e.g., microelectronics, computer systems)
- Battlefield simulations (blue army vs. red army)
- Ecological systems
- Manufacturing systems
- Logistics

Focus is typically on planning, system design

Simulations may take a long time to run

Applications (2): On-Line Decision Aids



Simulation tool is used for fast analysis of alternate courses of action in time-critical situations

- Initialize simulation from situation database
- Faster-than-real-time execution to evaluate effect of decisions

Applications: air traffic control, battle management

Simulation results may be needed in only seconds

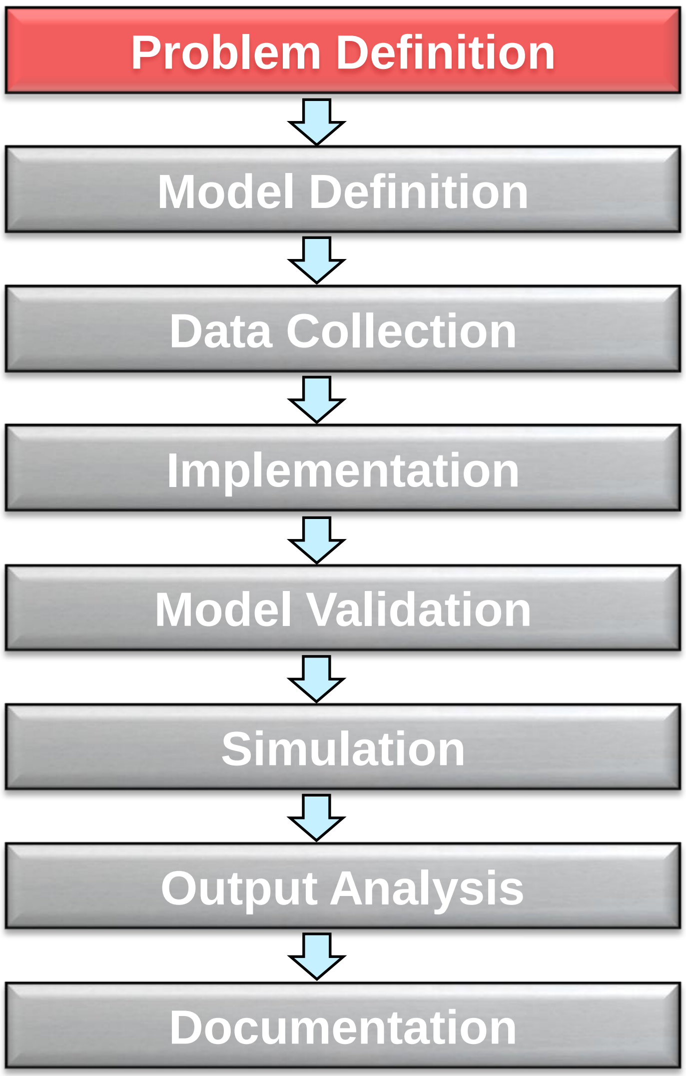
Applications (3): Virtual Environments

Uses: training (e.g., military, medicine, emergency planning), entertainment, social interaction?

Simulations are often used in virtual environments (human-in-the-loop) to create dynamic computer generated entities

- Adversaries and helpers in video games
- Defense: Computer generated forces (CGF)
  - Automated forces
  - Semi-automated forces
- Physical phenomena
  - Trajectory of projectiles
  - Buildings “blowing up”
  - Environmental effects on environment (e.g., rain washing out terrain)

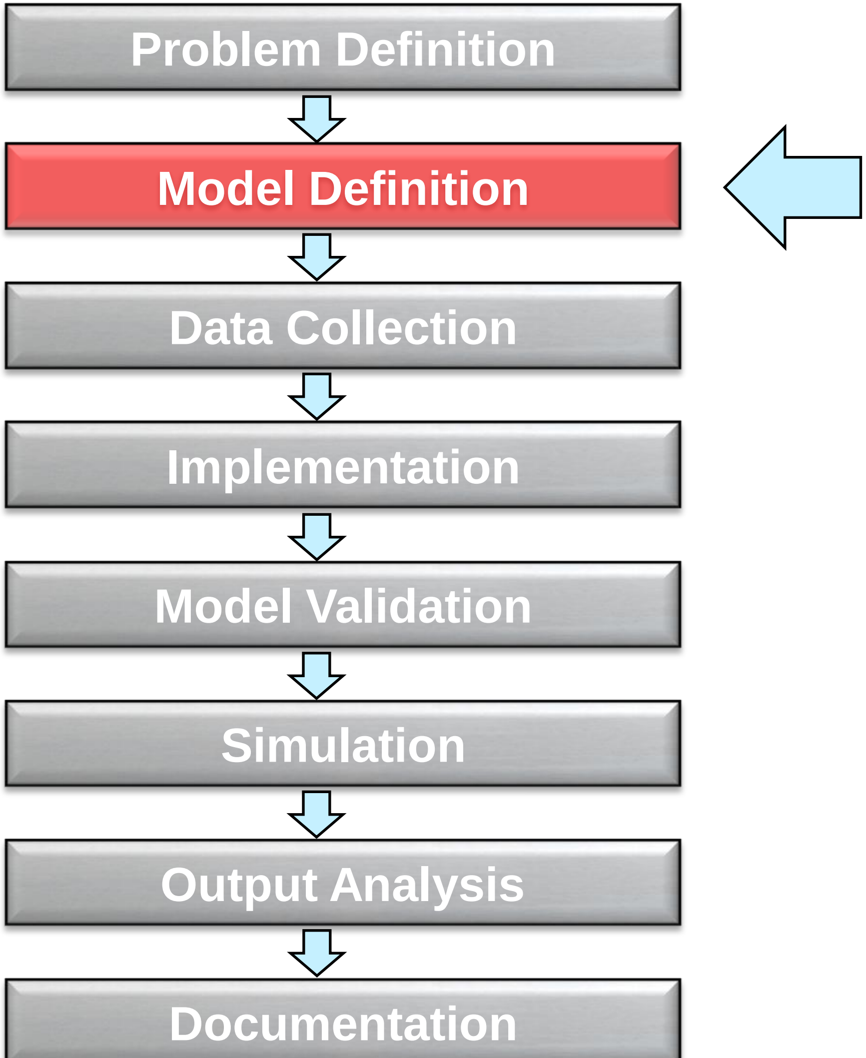
Typical Workflow – Problem Definition



Problem definition:

- What do I want to show?
  - Feasibility study
  - Performance study
  - Occurrence of specific phenomenon
- What is the desired complexity?
- How can I show it?
  - What are the inputs of the system?
  - What are the outputs?
  - What can I measure; what is inaccessible?
  - What may change, what will remain constant?
- Which parameters are important and which are not?

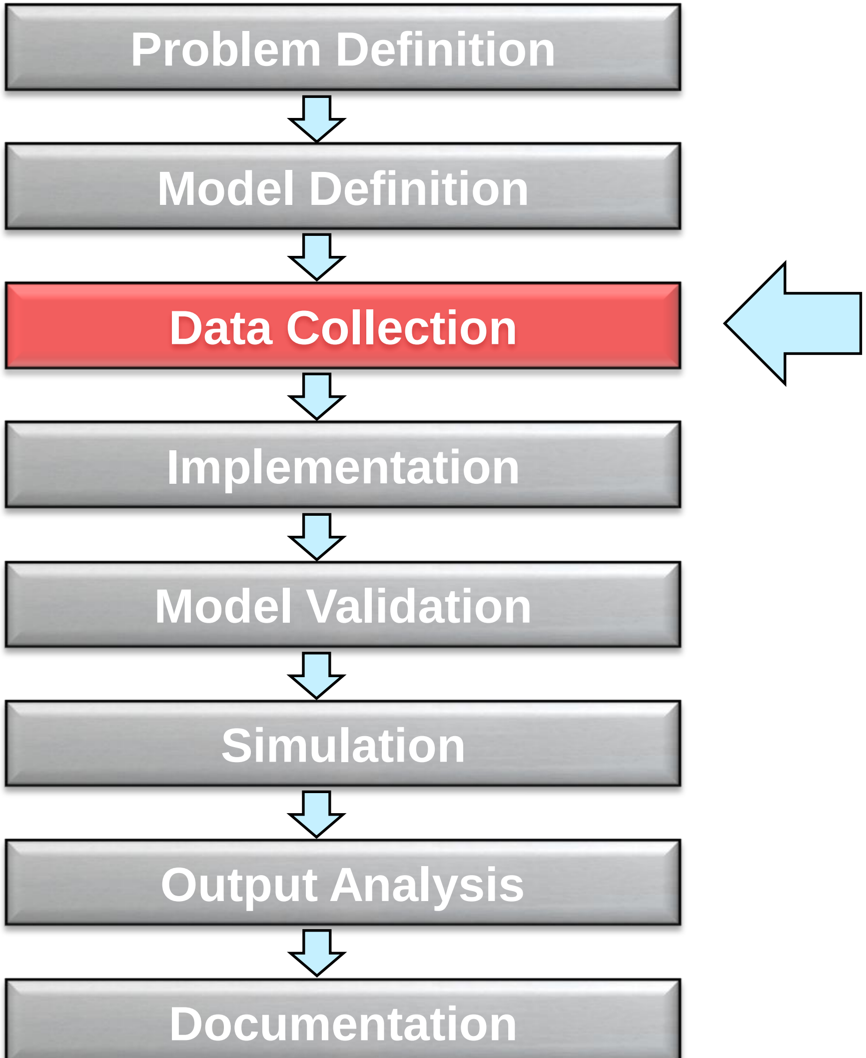
Typical Workflow – Model Definition



Model definition:

- Gain insight: How does the system behave?
- What is relevant for the model and what can be left out?
  - Level of detail
  - Processing power and memory consumption
- Identify the optimal model type (discrete/continuous, stochastic/discrete)
- Take advantage from a-priori knowledge

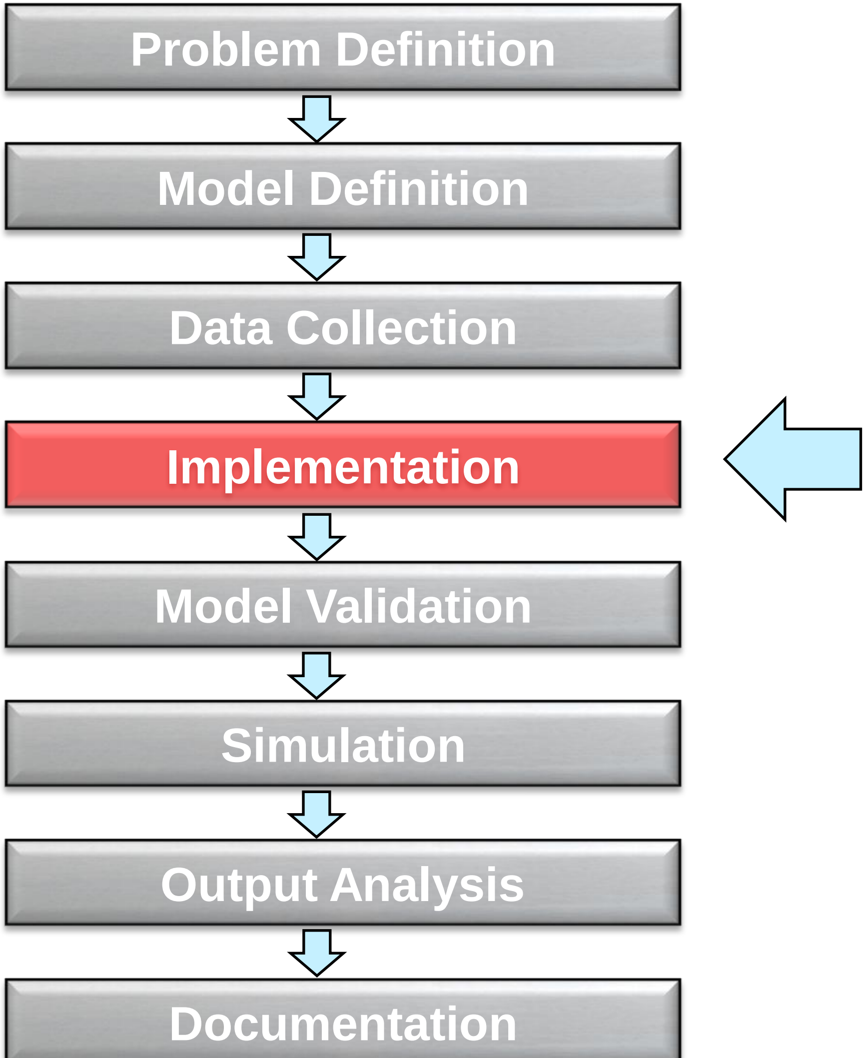
Typical Workflow – Data Collection



Data collection:

- Collect measurement data for validation
- Identification of input parameters
  - Configuration
  - System states
- Estimation of initialization parameters
- Evaluate and analyze output of the system

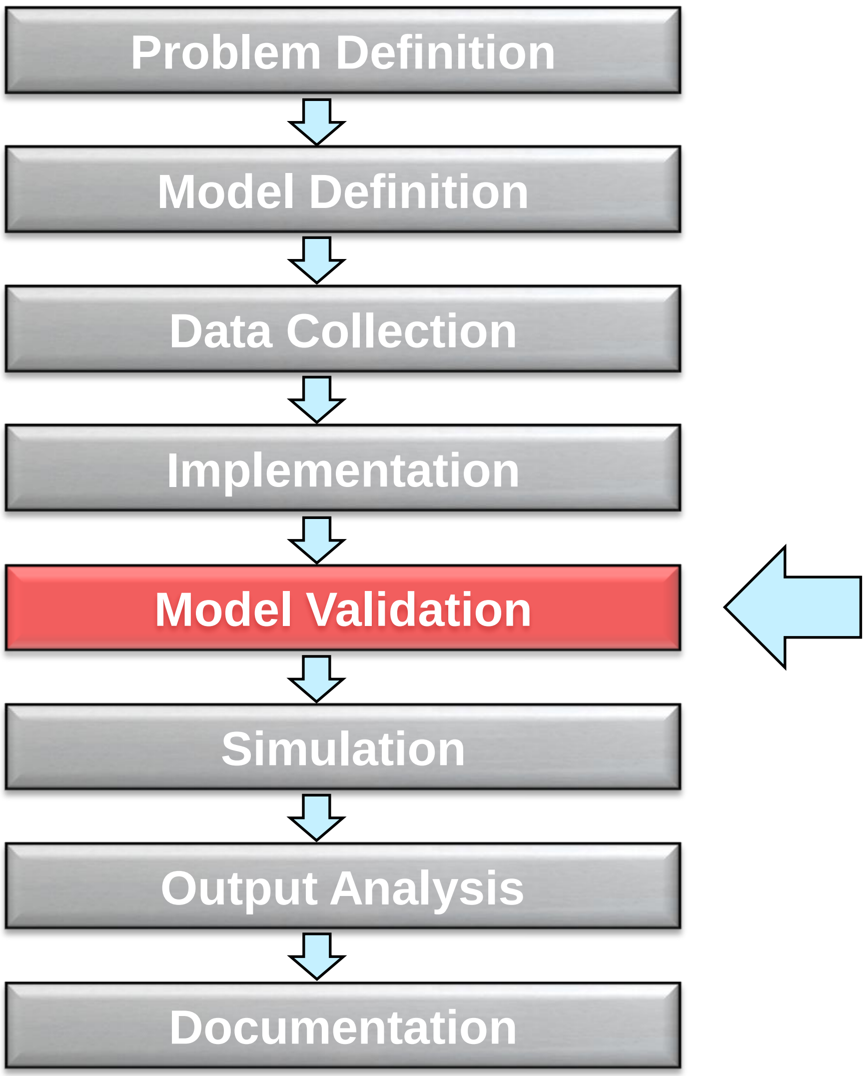
Typical Workflow - Implementation



Implementation:

- Implement the model such that considers all relevant parameters
- Always design your program with respect to reusability and extensibility
  - Note that implementation of a simulation is an iterative process
  - Models are usually extended
  - System states
- Select a suitable programming language:
  - Object oriented languages have proven to be very efficient (c.f. Simula)
  - Make use APIs for simulation
- Use a specialized simulation software if possible
  - ns-2, OMNET++, OPNET…

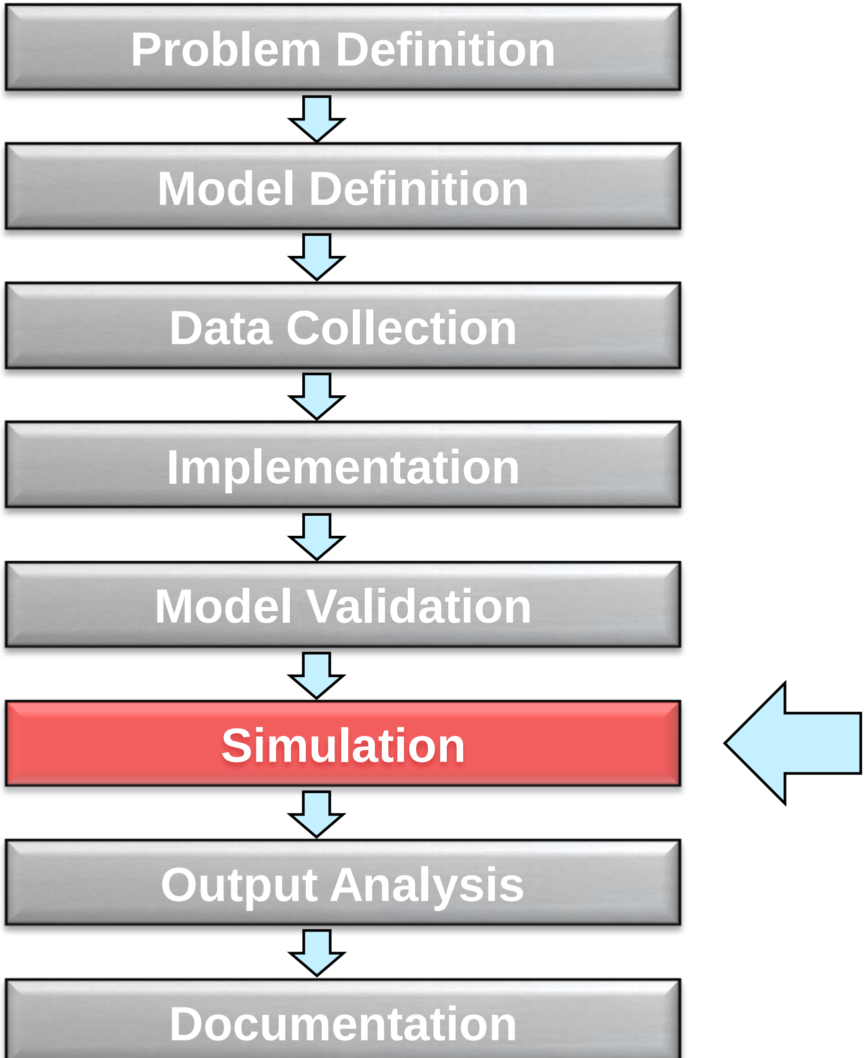
Typical Workflow – Model Validation



Model validation:

- Usual approach: Compare real data vs. simulation output
- Does the behavior of the simulation match with the real system?
- Identify the validity range of the simulation
- Can the simulation results be trusted?
- Validation loops:
  - Theoretical validation: Does it make sense? (steps 2 and 3)
  - Debugging: Is it correctly implemented? (steps 4–6)
  - Practical validation: Does it do the right things? (steps 2–6)
- Validation consumes a lot of time!

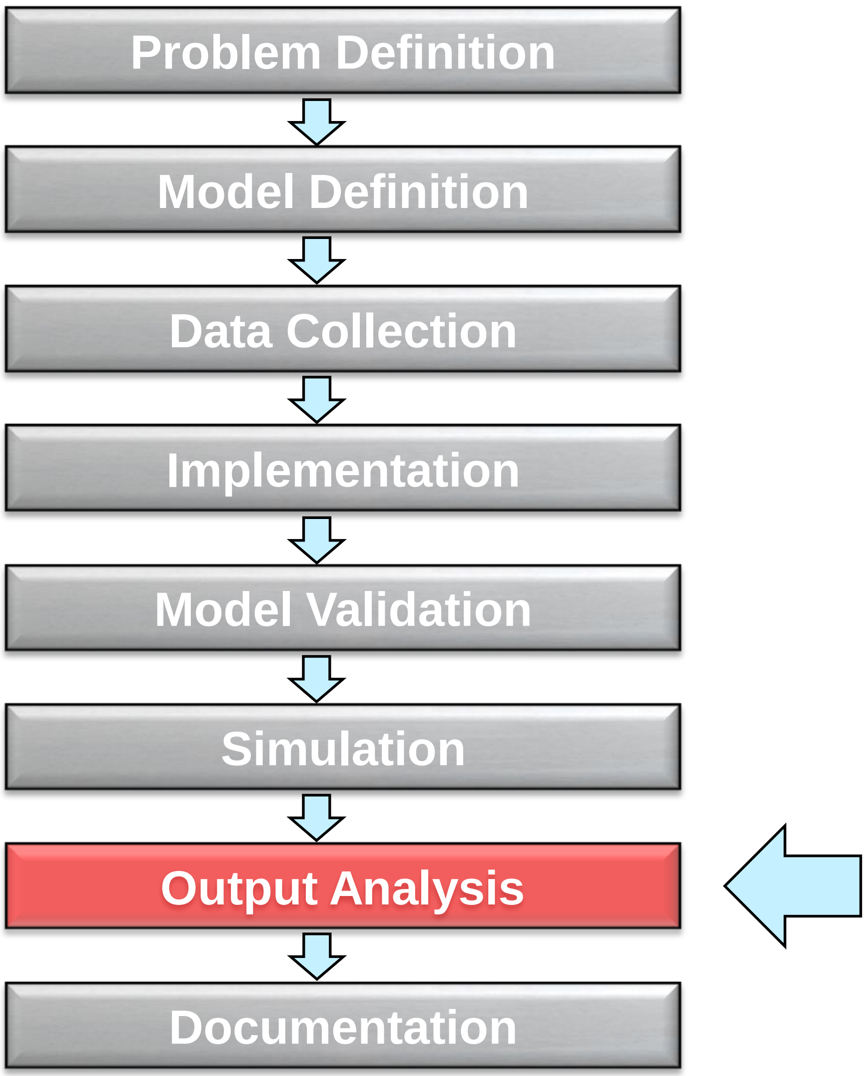
Typical Workflow - Simulation



Simulation:

- Input parameters are varied to evaluate their impact on the simulated system
- Identify characteristics of the simulation (duration of transient phase, simulation duration…)
- Determine required number of simulation runs and simulation duration
- Choose simulation parameters with respect to required hardware resources (processing power and memory consumption)

Typical Workflow – Output Analysis and Evaluation



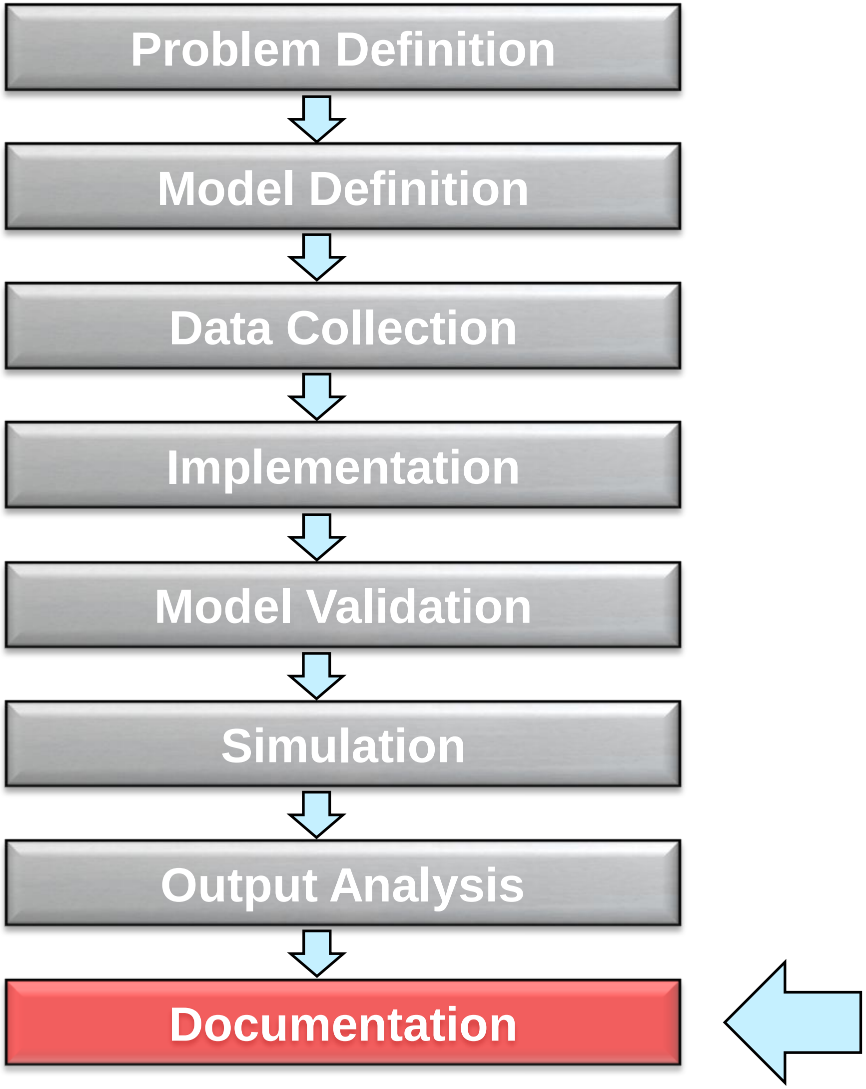
Output analysis and evaluation:

- Collected data is analyzed and interpreted:
  - Subjective evaluation
  - Objective evaluation
- Consider the impact of simplified assumptions of the simulation model
- Typical questions:
  - Comparison of systems and protocols
  - What-if-analysis

    How do changes of input parameters affect the performance parameter?
  - What-if-Achieve analysis

    Which costs/hardware is required to achieve a certain system performance?

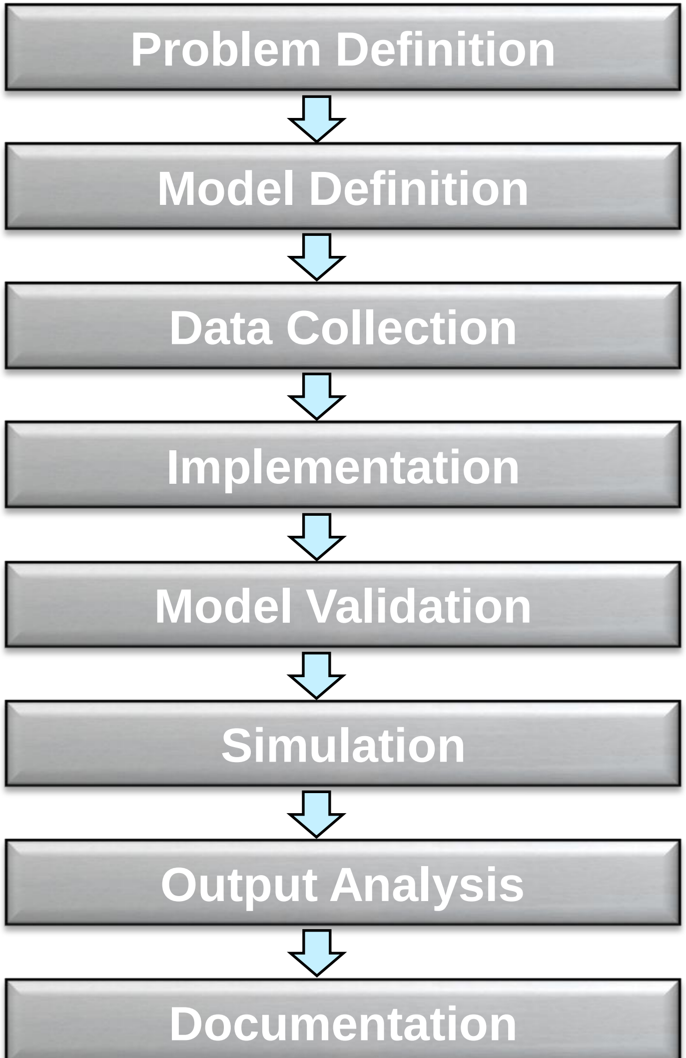
Typical Workflow - Documentation



Documentation:

- Documentation should be done in during all phases of the workflow
- Documentation assures reusability of the model
- Provide detailed model descriptions, especially simplifications which could lead to unnatural/unwanted effects
- Document the setup of the simulation
- Describe the applied evaluation methods
  - Error estimation
  - Confidence intervals

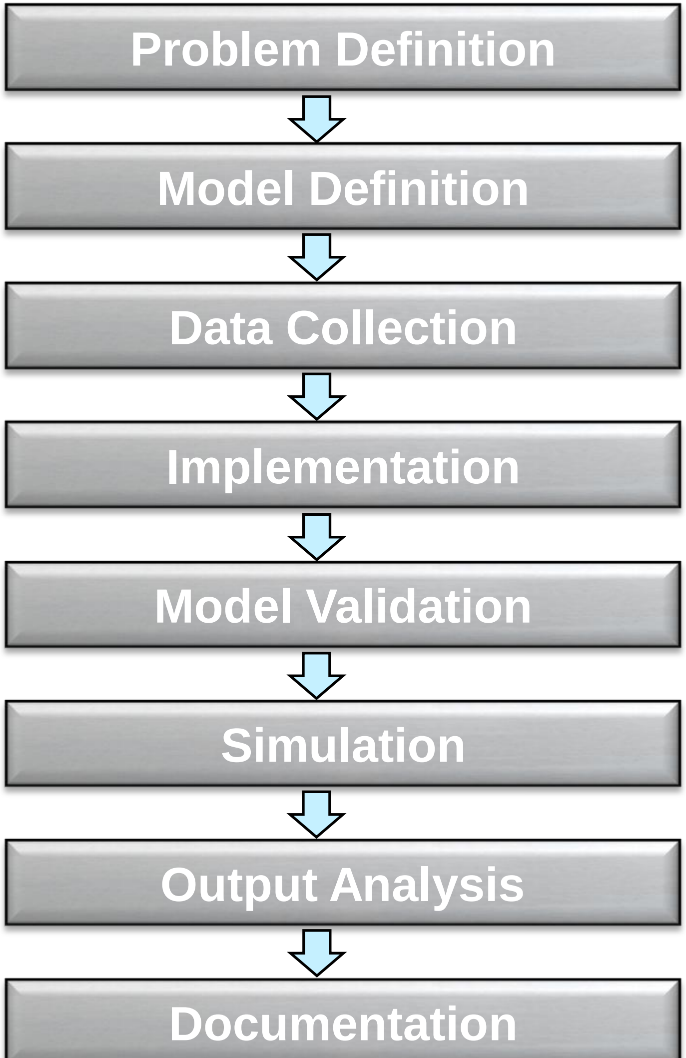
Typical Workflow – Typical Errors



- No clear understanding of the system
  - Impact of input parameters
  - Interactive behavior
  - Problem not identified
- No clear understanding of the simulation
  - Initialization, transient phase
  - Incomplete input parameters
- Inadequate abstraction level of the simulation

  Wrong results / slow simulation
- Simulation model invalid
  - Wrong assumptions lead to artifacts
  - Synchronized start time
- Analysis not correct
  - Small number of simulation runs

Typical Workflow – General Issues



- What do I want to show?
- How can I show it?
- What output do I need?
- What input do I want to try out?
- Usually infeasible many possible input patterns
- Selection is required
- Experiment planning and/or factorial design
- Often an iterative process
- Always distrust your results and model
- Recapitulate all previous steps if you are in doubt about the correctness of your simulation