Performance Analysis with POOSL

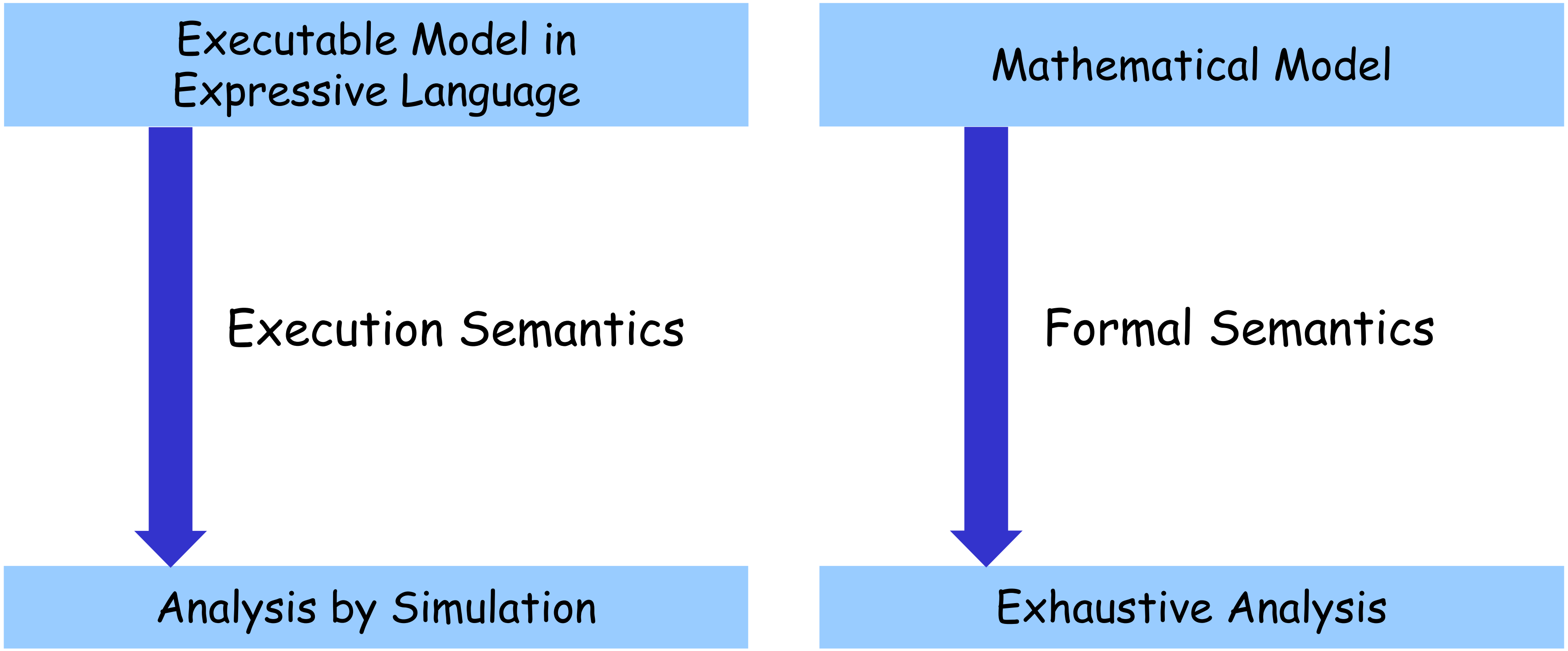
Bart Theelen (B.D.Theelen@tue.nl)
Jeroen Voeten (J.P.M.Voeten@tue.nl)

Department of Electrical Engineering
Electronic Systems

"Nothing is more simple than greatness;
indeed, to be simple is to be great"
Ralph Waldo Emerson (1803 - 1882)

Performance Modelling and Analysis

Design Practice based on | Formal Methods

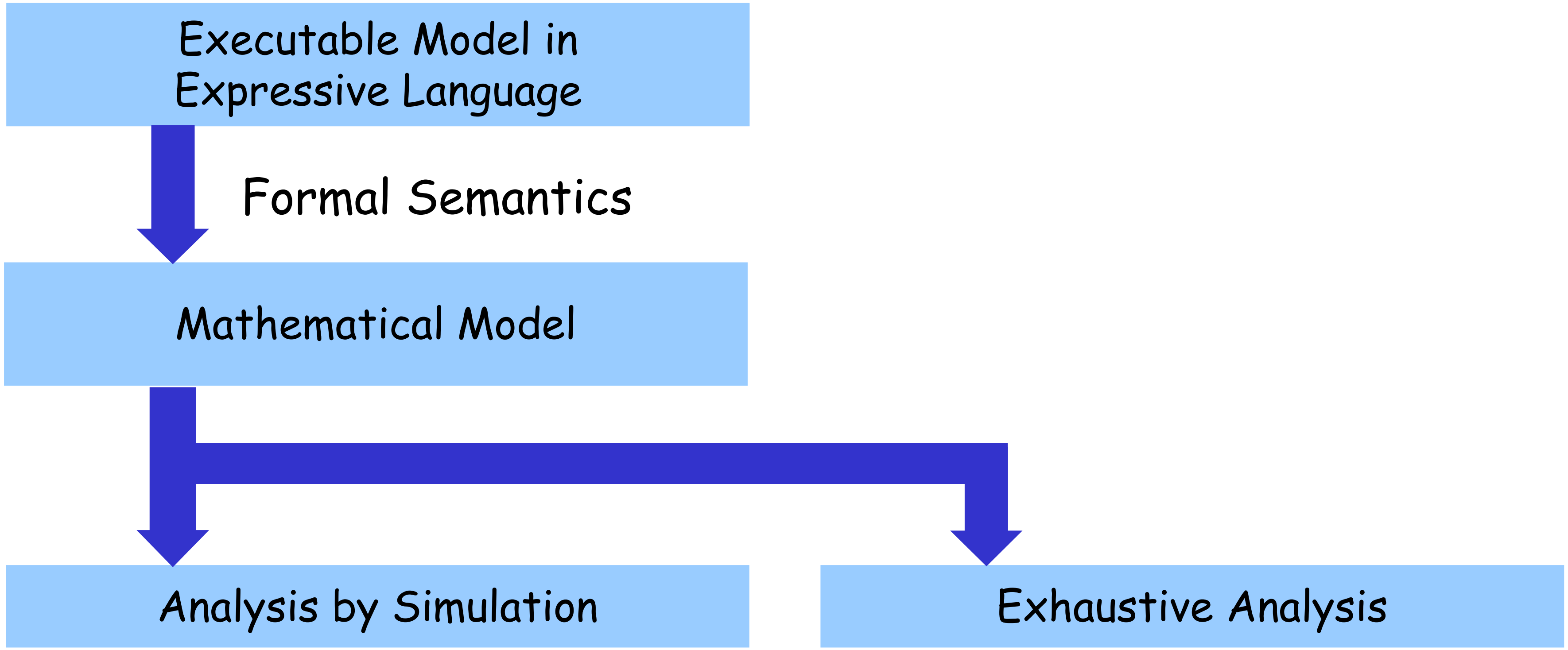


- Simulation-Based
  - Modelling convenience
  - Statistical analysis
  - Accuracy of results uncertain

- Exhaustive
  - Certainty about analysis results
  - Mathematical analysis
  - Difficult to obtain adequate models

3 Performance Modelling and Analysis with POOSL

Design Practice based on Formal Methods



- Simulation-Based
  - Modelling convenience
  - Statistical analysis
  - Accuracy of results uncertain
- Exhaustive
  - Certainty about analysis results
  - Mathematical analysis
  - Difficult to obtain adequate models

Performance Analysis Problems

- Hard real-time applications (satisfaction of requirements)
  - Throughput
  - End-to-end latency/delay
- Firm/soft real-time applications (satisfaction of requirements)
  - Throughput
  - End-to-end latency/delay and jitter
  - Deadline miss probability
- Platforms (bottleneck identification)
  - Average processor utilisation
  - Maximum, average of and variance in communication resource utilisation
  - Maximum, average of and variance in memory occupation
  - Peak and nominal power consumption

Fundamental problem:
No guarantees on
accuracy possible

	Worst/Best Case	Average Case
Exhaustive	I	II
Simulation-Based	III	IV

Assuming model
is adequate, still:

Requires indication of accuracy of results if approximation

Requires indication of accuracy of results

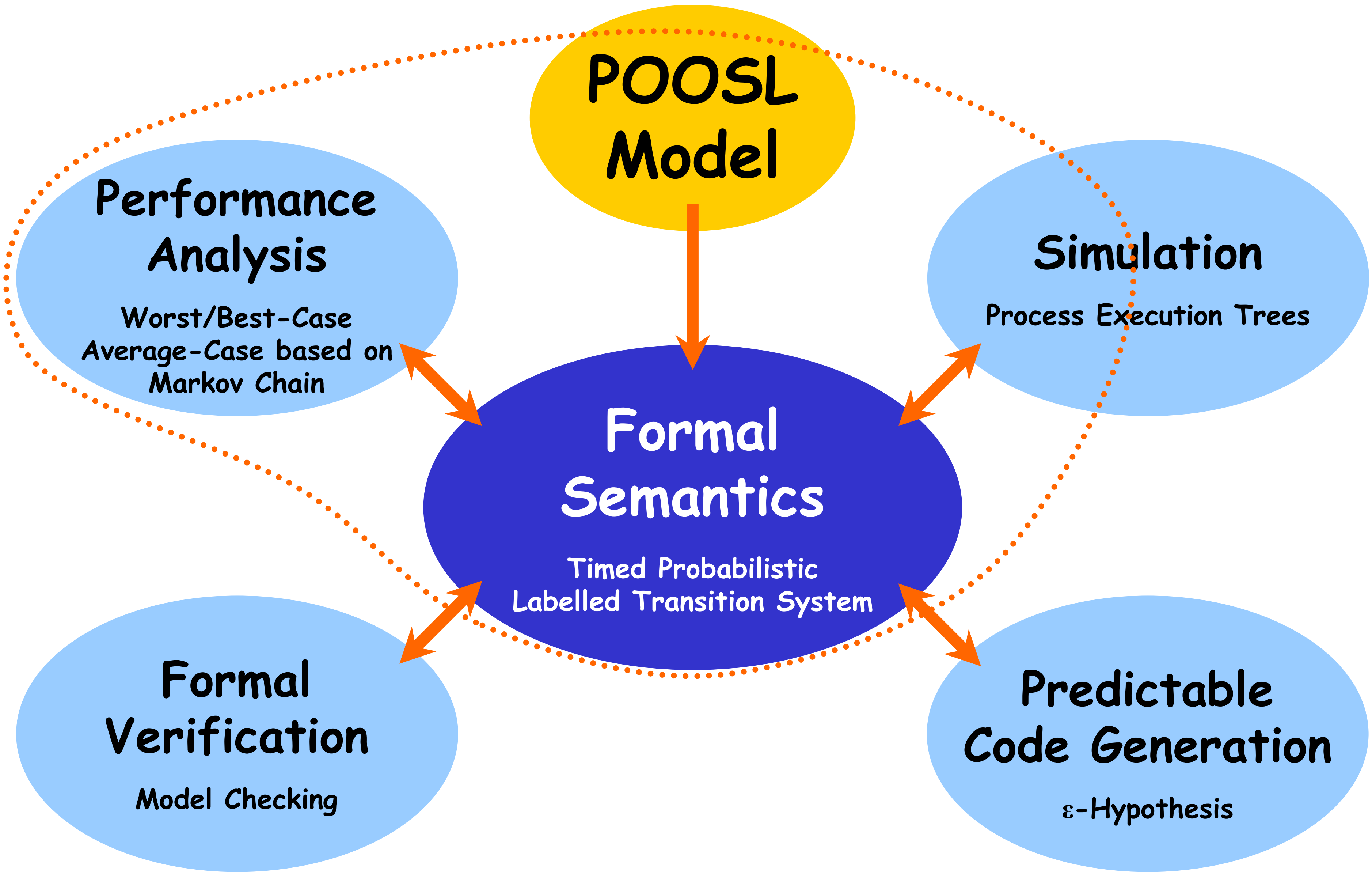
SHE Methodology: Overview

- SHE = Software/Hardware Engineering
- Modelling Languages
  - UML profile for SHE
  - Parallel Object-Oriented Specification Language
- Techniques
  - Simulation
  - Performance Analysis } Exhaustive and Simulation-based
  - Formal Verification } Exhaustive and Simulation-based
  - Code Generation
- Methods/Guidelines
  - Object-Oriented Analysis
  - Model Validation
  - Modelling Styles
  - Modelling Patterns
  - Design-Space Exploration
- Tools
  - SHESim
  - Rotalumis / Rotalumis-RT

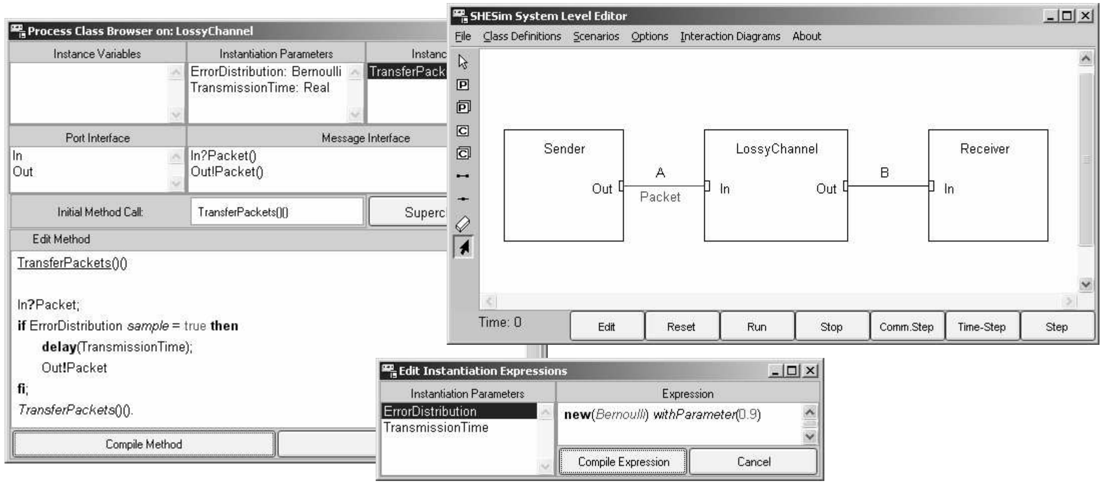
POOSL

- POOSL = Parallel Object-Oriented Specification Language
- Example of new generation of languages for system-level design
  - Bridge gap between industrial practice and formal methods
- Expressive
  - Asynchronous Concurrency
  - Synchronous Message Passing
  - Object-Oriented Data
  - Real-time and Stochasticity
  - Dynamic Process Creation
  - …
- Formal (Mathematical) Semantics
  - Probabilistic real-time extension of process algebra CCS
  - Traditional object-oriented programming languages (Java, Smalltalk)
- Executable

POOSL

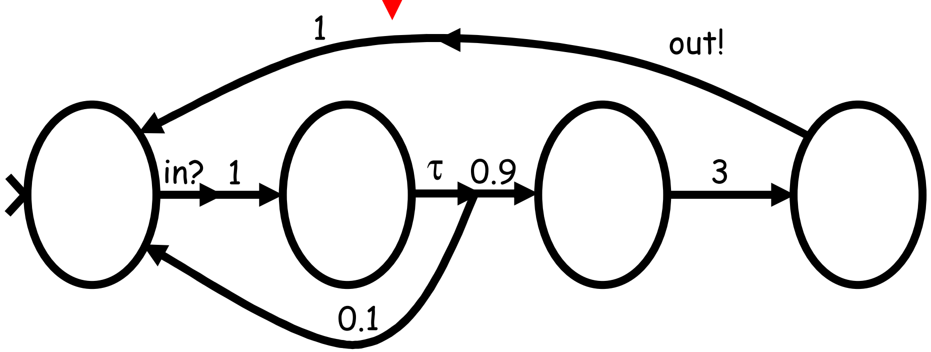

POOSL Model
Performance Analysis
Worst/Best-Case
Average-Case based on Markov Chain
Simulation
Process Execution Trees
Formal Semantics
Timed Probabilistic Labelled Transition System
Formal Verification
Model Checking
Predictable Code Generation
ε-Hypothesis

Example: Average-Case Performance Analysis



Understandable POOSL Model

Formal Semantics



Timed Probabilistic Labelled Transition System

Discrete-Time Markov Chain & Reward Structure

9 Features, Assumptions and Limitations

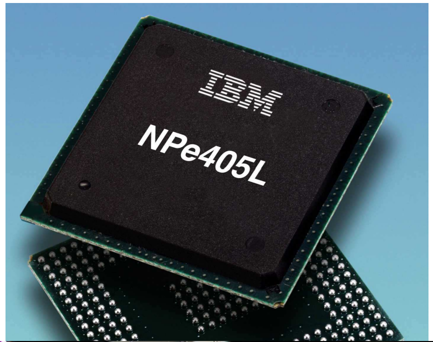
- Usual assumption: model is adequate

	Worst/Best Case Metrics	Average Case Metrics
Exhaustive	Exact Results Sample space of rewards has maximum/minimum Feasible if state space is small	Exact Results Markov chain is ergodic (any distribution allowed) Feasible if state space is small
Simulation-Based	Sample space of rewards has maximum/minimum Estimation results - no guarantees on accuracy of results	Markov chain is ergodic (any distribution allowed) Estimation results + bound on accuracy of results (for all metrics)

- No tools (yet) for exhaustive analysis

Application Domains / Industrial Case Studies

- Telecommunication Systems
  - Network Processor (IBM Research Laboratory)
  - High-Speed Packet-Switch (IBM Research Laboratory)
  - Internet Router (Alcatel Bell)
  - Data Flow System (Alcatel Bell)
  - DECT System for Hearing-Impaired Students (TNO Industrial Technology)
  - Intel IXP1200 Network Processor (University of Limerick)



- Real-Time Control Systems
  - MA3 System (TNO Industrial Technology)
  - Printer Controller System (Océ Technologies)
  - Wafer Stepper Subsystem (ASML)

- Multi-Media Systems
  - PiP TV Application (Philips Research Laboratory)
  - Design Space Exploration for a Car Navigation System (Siemens VDO)

11 Conclusions

- Strengths
  - Part of complete design methodology
    - Modeling (several modelling patterns and library components available)
    - Analysis (formal verification & worst/best-case, average-case performance analysis)
    - Synthesis
  - Based on formal modelling language POOSL
    - Intuitive (short learning curve)
    - Expressive (many models of computation, e.g., control, data flow, queuing, …)
    - Establishes link between formal methods and industrial practice
  - Scalable
  - Applicable to various application domains
    - Telecommunication systems
    - Real-time control systems
    - Consumer electronics / multi-media systems
    - High-tech systems

- Limitations
  - Limited possibilities for exhaustive analysis due to state-space explosion
  - No tools (yet) for exhaustive analysis

www.es.ele.tue.nl/poosl

Abstraction vs Adequacy

- The hardest part of system-level design is making adequate abstractions when developing models of design alternatives
- Model-based analysis allows for answering specific questions
- Two properties of models:
  - Abstraction = discarding details that are irrelevant for answering questions
    - Necessary: Many implementation details are (still) unknown
    - Desirable: Allows postponing design decisions on details

    Improvement in analysis speed is merely a nice positive effect
  - Adequate = including all aspects that are relevant for answering questions
    - Model represents system properly with respect to aspects relevant for questions
- Abstraction and adequacy are conflicting objectives
  - We want representative results without taking all details into account
- Why is it the most difficult part?
  - Adequacy of a model can only be confirmed after realising the system
- Any method should include techniques for validating adequacy of models

13 Adequacy vs Accuracy

- Adequacy is property of model
  - We all assume that model is adequate after certain modelling effort
- Accuracy is property of result
  - Exhaustive approaches give exact results -> 100% accurate
  - Simulation-based approaches may not give 100% accurate results
    - Any simulation result should be accompanied by bound on error
- An inadequate model can give perfectly accurate results
  - Example: Queuing network (exhaustive analysis), where the distributions in the represented system are not exponential
- A perfectly adequate model can give inaccurate results
  - Example: Simulation-based analysis of worst/best case

Exhaustive vs Simulation-Based

- In case a tool relies on a modelling language for which a rigorous framework to compute performance metrics (exhaustive approach) is missing, simulation-based estimation with this tool cannot lead to credible results

- Rigorous framework
  1. Makes models amenable to analysis techniques
  2. Allows for unambiguous execution of models

- Analytical computation requires satisfaction of 1
  1. Is required for getting results properly

- Simulation-based estimation requires satisfaction of 1 and 2
  1. Is required for getting results properly and for analysing their accuracy
  2. Is required for guaranteeing unambiguous results