Performance Modeling of Computer Systems and Networks

Prof. Vittoria de Nitto Personè

Simulation introduction

Università degli studi di Roma Tor Vergata
Department of Civil Engineering and Computer Science Engineering

1

Performance evaluation techniques

Computational and mathematical techniques to *model*, *simulate* and *analyze* the performance of *stochastic systems*

Modeling: conceptual framework describing a system

Simulate: perform experiments using computer implementation of the model

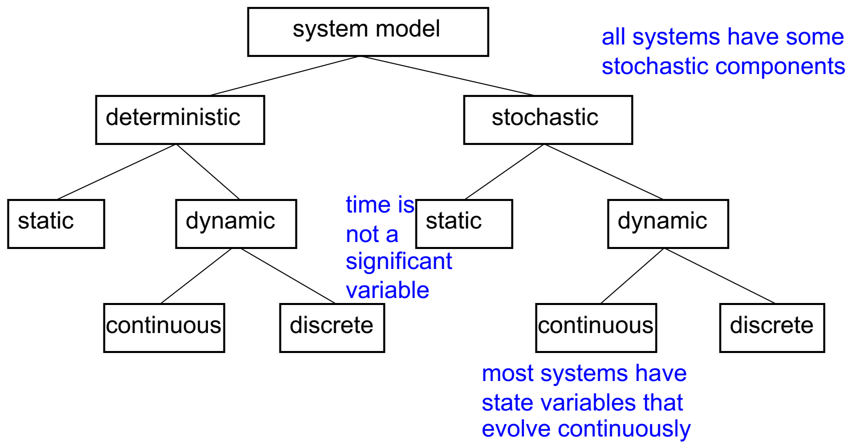
Analyze: draw conclusions from output

Simulation models

Analytical models

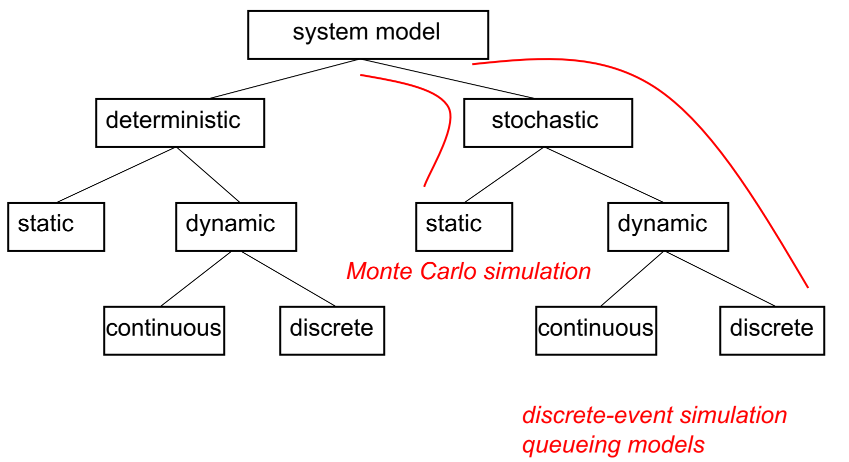
2

Characterizing a model



3

Characterizing a model



4

Algorithm 1.1: how to develop a model

1. Goals and objectives e.g. Boolean decisions
   Numeric decisions **UNBIASED**
2. *Conceptual* model (cm)
3. Convert cm into a *specification* model (sm)
4. Convert sm into a *computational* model (cptm)
5. Verify
6. Validate

Three Model Levels

i. Conceptual
   - very high level
   - which are the state variables, how they are related, which can be ignored and which not

ii. Specification
   - On paper
   - May involve equations, pseudocode, etc.
   - How will the model receive input?
     - collecting and statistically analyzing data
     - using representative stochastic models

iii. Computational
   - A computer program
   - General-purpose PL or simulation language?

Verification vs. Validation

5. Verification
   - Computational model should be consistent with specification model
   - Did we build the model right?
6. Validation
   - Computational model should be consistent with the system being analyzed
   - Did we build the right model?
   - Can an expert distinguish simulation output from system output?

Algorithm 1.1: how to develop a model

1. Goals and objectives e.g. Boolean decisions
   Numeric decisions **UNBIASED**
2. *Conceptual* model (cm)
3. Convert cm into a *specification* model (sm)
4. Convert sm into a *computational* model (cptm)
5. Verify
6. Validate

Typically an iterative process

Algorithm 1.1: observations

1. Goals
2. Conceptual model
3. Specification model
4. Computational model
5. Verify
6. Validate

- Make each model as simple as possible:
  - Never simpler
  - Do not ignore relevant characteristics
  - Do not include extraneous characteristics
- Model development is not sequential
  - Steps are often iterated
  - For teams, steps may be in parallel
  - Do not merge verification and validation
- Develop models at three levels
  - Think a little, program a lot (and poorly);
  - Think a lot, program a little (and well).

~~1. Goals~~
~~2. Conceptual model~~
~~3. Specification model~~
~~4.~~ 1. Computational model
~~5.~~ 2. Verify
~~6.~~ 3. Validate

Certainly produce large, inefficient, unstructured cm that CANNOT BE VALIDATED

Algorithm 1.2: using the resulting model

7. Design simulations experiments
   - What parameters should be varied?
   - perhaps many combinatoric possibilities
8. Make production runs
   - Record initial conditions, input parameters
   - Record statistical output
9. Analyze the output
   - Random components → statistical analysis (means, standard deviations, percentiles, histograms etc.)
10. Make decisions
    - The step9 results drive the decisions → actions
    - Simulation should be able to correctly predict the outcome of these actions (→ further refinements)
11. Document the results
    - summarize the gained insights in specific observations and conjectures useful for subsequent similar system models

Machine Shop Model

- 150 identical machines:
  - Operate continuously, 8 hr/day, 250 days/yr
  - Operate independently
  - Repaired in the order of failure
  - Income: 50,00 €/hr of operation
- Service technicians:
  - 2-year contract at 60.000,00 €/yr
  - Each works 230 8-hr days/yr

Machine Shop Model

1. Goals
2. Conceptual model
3. Specification model
4. Computational model
5. Verify
6. Validate

- How many service technicians should be hired to maximize the profit?

  Extreme solutions: just 1 technician
  → minimizes service-techn overhead
  → large down-times
  → loss of income

  1 technician for each machine
  → huge service-techn overhead
  → minimum down-times
  → maximizes income

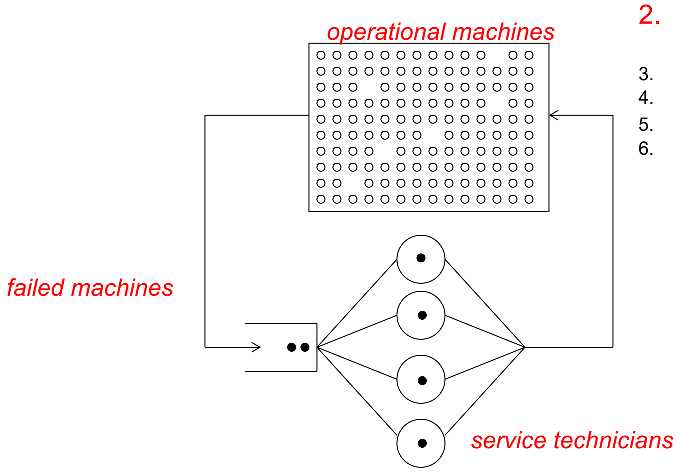
Machine Shop Model

1. Goals
2. Conceptual model
3. Specification model
4. Computational model
5. Verify
6. Validate

- State of each machine (failed, operational)
- State of each techn (busy, idle)
- Provides a high-level description of the system at any time

Machine Shop Model

1. Goals
2. Conceptual model
3. Specification model
4. Computational model
5. Verify
6. Validate



model development
example

Machine Shop Model

1. Goals
2. Conceptual model
3. **Specification model**
4. Computational model
5. Verify
6. Validate

- What is known about time between failures? Are the failures random?
- What is the distribution of the repair times?
- How will time evolution be simulated?

15

model development
example

Machine Shop Model

1. Goals
2. Conceptual model
3. Specification model
4. **Computational model**
5. Verify
6. Validate

- It should include:
  - Simulation clock data structure
  - «Queue» of failed machines
  - «Queue» of available technicians
  - performance characterization (structures to collect statistical data)

16

model development
example

Machine Shop Model

1. Goals
2. Conceptual model
3. Specification model
4. Computational model
5. **Verify**
6. Validate

- Software engineering activity
- Usually done via extensive testing

Prof. Vittoria de Nitto Personè 17

17

model development
example

Machine Shop Model

1. Goals
2. Conceptual model
3. Specification model
4. Computational model
5. Verify
6. **Validate**

the validation step allows to verify if the cptm is a "good approximation" of the actual machine shop

- If operational, compare against the real thing

- otherwise → use *consistency checks*

e.g. as the n. of technies grows, the average n. of fault machines decreases

as the mean service time grows, the average n. of fault machines grows too

Prof. Vittoria de Nitto Personè 18

18

Machine Shop Model

7. Experiments design
8. Runs production
9. Output analysis
10. Decisional phase
11. Results documentation

- Find the optimal number of technies to maximize profit
- Initial conditions (e.g. are all machines initially operational?)
- For a fixed n. of service technies, how many replications are required to reduce the natural sampling variability in the output statistics to an acceptable level?

Machine Shop Model

7. Experiments design
8. Runs production
9. Output analysis
10. Decisional phase
11. Results documentation

- If many runs are made,
  management of the output results becomes an issue
  → avoid to archive "raw date"
  simulation advantage: experiments can always be reproduced

Machine Shop Model

- The statistical analysis (sa) of sim output often is more difficult than classical sa
  → *dependent (/correlated)* observations
  e.g. if the current n. of failed machines is observed each hour, consecutive observations will be found positively correlated → both below or above the mean n. of failed machines
- ATTENTION to erroneous conclusions

Machine Shop Model

- A graphical display of profit versus the number of service technies yields both the optimal n. of technies and a measure of how sensitive the profit is to variations of this n. (cost)
- *Decision policy not violating any external constraint*

Simulation studies
example

Machine Shop Model

7. Experiments design
8. Runs production
9. Output analysis
10. Decisional phase
11. Results documentation

- System diagram
- Assumptions about failure and repair rates
- Description of specification model
- Software
- Tables and figures of output
- Description of output analysis

Advantages of the sim study:
can provide valuable insights about system features and component interactions otherwise not achievable

Prof. Vittoria de Nitto Personè 23

23

terminology

- *Model* / *simulation* (noun)
  - Model can be used with respect to conceptual, specification, or computational levels and for both analytical and simulation techniques
  - Simulation is frequently used to refer to the computational model (program), it is rarely used to describe the conceptual or specification model
- *Model* / *simulate* (verb)
  - To model can refer to development of the levels
  - To simulate refers to the computational activity
- ATTENTION do not confuse *verify* with *validate*

Prof. Vittoria de Nitto Personè 24

24

Exercises

- Ex 1.1.2 and Ex 1.1.3 on p.11 from textbook