

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

Copyright © Vittoria de Nitto Personè, 2021
<https://creativecommons.org/licenses/by-nc-nd/4.0/>



1

Simulation introduction

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

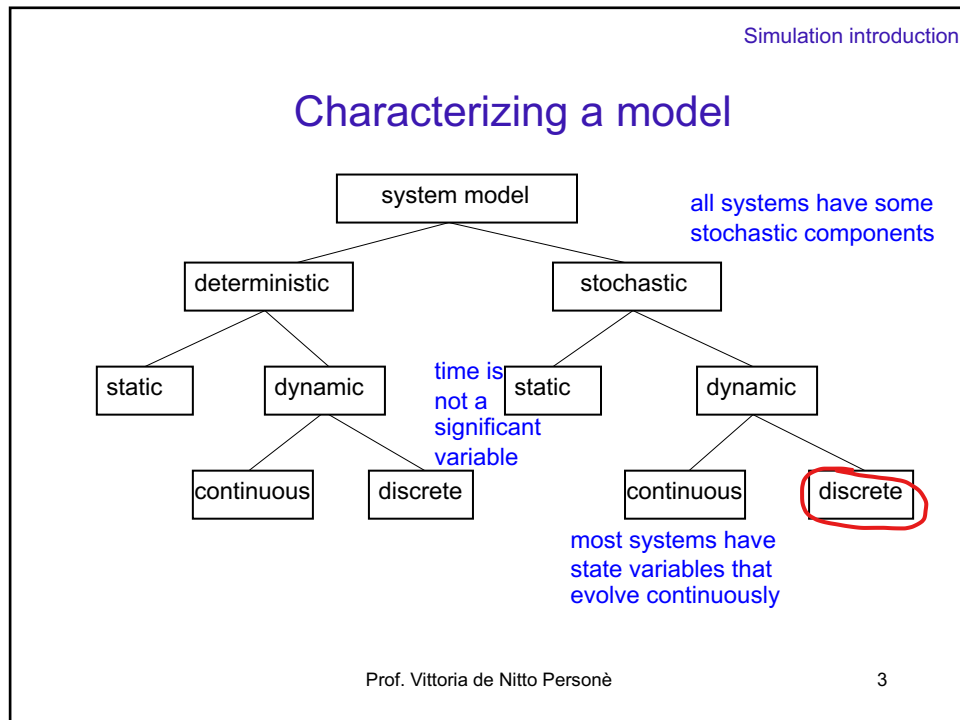
Prof. Vittoria de Nitto Personè

2

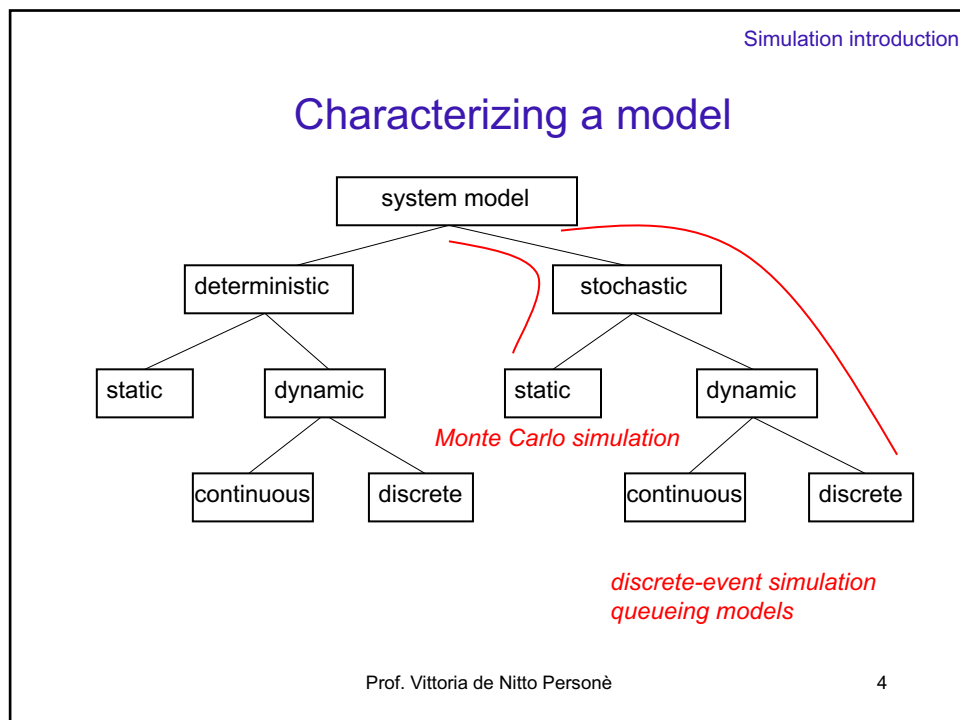
2

abbiamo visto che parliamo di un insieme di tecniche, da analitici alla simulazione, che hanno l'obiettivo di analizzare le prestazioni, punto di arrivo di tutti gli studi.
Vediamo la parte SIMULATIVA.

Simulo modello, quindi devo definire tale modello. Può essere deterministico, stocastico, statico, dinamico, continuo, discreto. Tutti i sistemi hanno almeno un componente stocastico. Normalmente ci interessiamo al caso 'dinamico' (statico tempo non rilevante). Quindi lo spazio può essere continuo o discreto, normalmente ci basiamo su modelli a coda DISCRETI.



3



4

- come sviluppo il modello?
0. individuo caso di studio, nella slide è 'scontato'.
1. obiettivi: devo aggiungere un server? (quesiti si/no), o quanti server mi servono (quesiti numerici).
Non devo mai pensare 'non so fare questa cosa, quindi me la acchitto in modo da non trattarla', non devo farmi condizionare.
2. modello concettuale cm : quali elementi che caratterizzano il sistema sono essenziali e sufficienti per poter studiare gli obiettivi del punto 1? step FONDAMENTALE.
3. converto 2. in 'modello delle specifiche sm', cioè comincio a caratterizzare flusso di arrivo, distribuzioni, cioè parametrizzo quel modello.

08/03/22

model development

Algorithm 1.1: how to develop a model

- Goals and objectives e.g. Boolean decisions
Numeric decisions **UNBIASED**
- Conceptual* model (cm)
- Convert cm into a *specification* model (sm)
- Convert sm into a *computational* model (cptm)
- Verify
- Validate

Prof. Vittoria de Nitto Personè

5

4. converto 3. in 'modello computazionale cptm', cioè inizio la simulazione, scrivo il programma.
5. VERIFICA, questo modello 4. è corretto rispetto alle specifiche?
6. VALIDAZIONE, stabilisco se il modello scelto è un buon modello per il sistema reale, non se è corretto oppure no, questo lo faccio con la verifica. Se c'è qualche errore, torno indietro al punto 2.

model development

Three Model Levels

- Conceptual**
 - very high level
 - which are the state variables, how they are related, which can be ignored and which not
- Specification**
 - On paper
 - May involve equations, pseudocode, etc.
 - How will the model receive input?
 - collecting and statistically analyzing data
 - using representative stochastic models
- Computational**
 - A computer program
 - General-purpose PL or simulation language?

Prof. Vittoria de Nitto Personè

6

- 6
- concettuale: molto astratto, richiede preparazione, devo capire variabili di stato (con definizione ben precise, fondamentali da capire bene), quali sono, come sono legate tra loro, quali posso ignorare. si fa 'su carta'.
 - specifiche: posso usare equazioni, pseudocodice, relazioni etc per trovare legami tra variabili di stato. Come si riceve l'input? analizzando dati e usando modelli stocastici.
 - computazionale: programma vero e proprio. Noi usiamo linguaggi general-purpose (C, Java, Python quelli che possiamo usare per il progetto).

Molto spesso si tende a confonderli, o pensare siano la stessa cosa. E' un grave errore.

model development

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?

Prof. Vittoria de Nitto Personè

7

7

- verifica: ho costruito il modello in modo corretto?

- validazione: ho costruito il modello corretto? un esperto può distinguere output reale da quello simulato?

model development

Algorithm 1.1: how to develop a model

1. Goals and objectives e.g. Boolean decisions **UNBIASED**
Numeric decisions
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

Prof. Vittoria de Nitto Personè

8

8

model development

Algorithm 1.1: observations

- 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. Computational model
5. Verify
6. Validate

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

Prof. Vittoria de Nitto Personè 9

9

Simulation studies

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

risultati vanno documentati, sintetizzati. Progetti ben documentati si prestano bene a studi futuri.

Prof. Vittoria de Nitto Personè 10

che parametri variare, capendo quando ha senso variarne solo uno alla volta, e quando no.

concetti di indipendenza e identicamente distribuiti non è detto che ci siano nelle simulazioni. queste sono alla base del punto 10

10

Abbiamo tante run di esecuzione (almeno 500), e quindi non posso variare tutto a caso. Simulazioni ripetibili, non devo 'mantenere' output di ogni osservazione. Fare tabelle excel giganti con tante simulazioni ed indici ha poco senso, a noi interessano le medie delle simulazioni. Mantengo statistiche di output, non tutti i valori.

Digressione su qualche progetto degli anni precedenti:

1. si individui sistema oggetto di studio: NON devo partire col modello, ma devo capire cosa voglio studiare. Devo descrivere caso di studio.

2. obiettivi dello studio

3. ORA costruisco il modello. Non posso presentare modello a rete di code al punto 1.

tra punto 3 e 4 c'è la progettazione degli esperimenti.

In alcuni casi non c'è fase stazionaria, ma a volte possiamo 'forzarla' perchè ci è indice di alcuni comportamenti limite. Non sempre è richiesta. E' diverso dal caso transiente.

08/03/22

"obiettivo dello studio è realizzare modello a code..." NON VA BENE, quello è il metodo. L'obiettivo era simulare il sistema giudiziaria.

buona idea far parallelismi: job = denunce, server = giudici, etc.. cioè a livello di astrazione.

spesso fasi del dominio complicate sono approssimate con single server.

model development
example

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

Prof. Vittoria de Nitto Personè

11

11

model development
example

Machine Shop Model

- 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

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

Prof. Vittoria de Nitto Personè

12

12

model development
example

Machine Shop Model

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

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

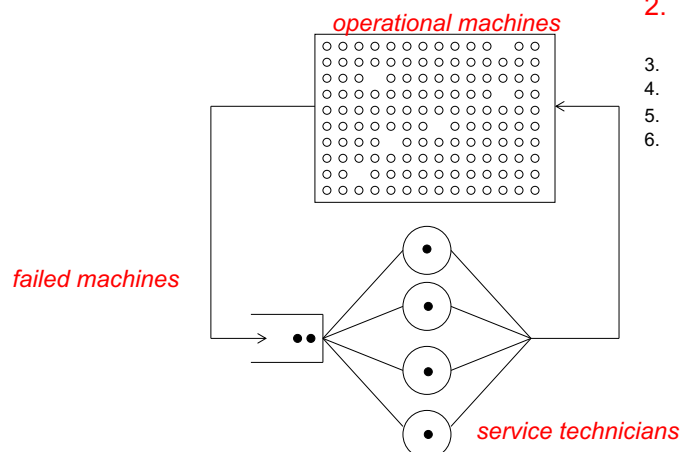
Prof. Vittoria de Nitto Personè

13

13

model development
example

Machine Shop Model



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

Prof. Vittoria de Nitto Personè

14

14

model development
example

Machine Shop Model

- 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?

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

Prof. Vittoria de Nitto Personè

15

15

model development
example

Machine Shop Model

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

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

Prof. Vittoria de Nitto Personè

16

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

- 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?

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

Machine Shop Model

- 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

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

Machine Shop Model

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

- 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

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

- 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*

Machine Shop Model

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

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

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

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*

Exercises

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

Prof. Vittoria de Nitto Personè

25