


## Performance Modeling of Computer Systems and Networks

*Prof. Vittoria de Nitto Personè*

### Next Event Simulation

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/>  (CC BY-NC-ND 4.0)

1

p.181 Discrete

DE simulation  
Next-Event Simulation

## Next-Event Simulation

5 concetti alla base della Next Event  
Next-event simulation is a more general approach to  
discrete-event simulation

- system state
- events
- simulation clock
- event scheduling eventi da schedulare in ordine di tempo
- event list lista degli eventi

Prof. Vittoria de Nitto Personè 2

2

Con la Discrete Event vista finora, l'evolvere del tempo non c'è. Qui simuliamo veramente il tempo, prima non c'era un simulatore temporale. Devo simularlo per davvero, non tutto insieme. L'idea è quella di avere un sistema più generale e quindi facilmente modellabile.

## Definitions and Terminology – State

The *state* of a system is a complete characterization of the system at an instance in time

Ci mette in condizione di capire come il sistema evolverà all'istante che segue, all'istante del prossimo evento.

Ricordiamoci di avere tre modelli, lo stato sarà per ciascuno di questi tre livelli.

## Algorithm 1.1: how to develop a model

1. Goals and objectives
2. *Conceptual model* (cm)
  - very high level
  - which are the state variables, how they are related, which can be ignored and which not
3. Convert cm into a *specification model* (sm)
  - important: collecting and statistically analyzing data to provide the input models that drive the simulation
4. Convert sm into a *computational model* (cptm)
5. Verification
  - Is cptm consistent with sm?
6. Validation
  - Is cptm consistent with the system being analyzed?
  - Can an expert distinguish simulation output from system output?

## Definitions and Terminology – State

- **Conceptual model:**  
abstract collection of variables and how they evolve over time
- **Specification model:**  
collection of mathematical variables together with logic and equations
- **Computational model:**  
collection of program variables systematically updated

devo caratterizzare sistema ad un istante di tempo e dire come evolve.

- **Example ssq:** the state is number of jobs in the node (a livello concettuale)
- **Example inventory system:** the state is current inventory level  
caratterizzo in un determinato tempo lo stato del sistema,  
posso anche dire come cambiava tale stato in presenza di un ordine

Prof. Vittoria de Nitto Personè

5

5

## Definitions and Terminology - Events

An **event** is an occurrence that may change the state of the system.

By definition, state cannot change except at an event time.

Each event has an associated **event type**.

- We can define **artificial events (do not change system state)** quasi sempre sono per raccolte dati
  - Statistically sample the state of the system
  - Schedule an event at a prescribed time  
(block arrival flow into the node, an inventory review without orders etc.)

esempio: voglio campionare le statistiche, oppure riporto il sistema allo stato iniziale.  
In quest ultimo caso sto cambiando lo 'stato', ma non è un evento che naturalmente cambia lo stato, è un evento forzato.

Prof. Vittoria de Nitto Personè

6

6

## Definitions and Terminology - Simulation Clock

The **simulation clock** represents the current value of simulated time

- Discrete-event simulations lack definitive simulated time  
As a result, it is difficult to generalize or embellish models

Nei modelli visti precedentemente (Discrete Event), avevamo in realtà due clock, legati ad arrivi e servizi. Questi due clock sono però scollegati, e quindi operazioni apparentemente semplici come la gestione del feedback, che richiede il loro legame, risultavano più complesse.

Altro esempio nella gestione dell'inventario:  
Lì ho solo un "clock" che va di settimana in settimana, quindi non posso vedere le singole richieste di approvvigionamento negli istanti precisi, ma vengono raggruppate alla fine della settimana. Per lo studio iniziale va bene, ma appena volessi introdurre una qualche metrica in più (richiesta in un certo delta di tempo) perdo precisione.

Prof. Vittoria de Nitto Personè

7

7

## Definitions and Terminology - Event Scheduling & Event List

### *scheduler*

- a *time-advance mechanism* avanzamento del tempo  
to guarantee that events occur in the correct order
- *next-event time advance* is typically used in discrete-event simulation "Next Event" perchè il tempo si sposta tra eventi, se il prossimo evento è tra 5 sec, vado direttamente su di lui.

### *event list*

- the data structure containing the time of next occurrence for each event type mantiene l'ordine in cui gli eventi devono accadere

To build a **next-event simulation**:

- construct a set of state variables definire set variabili stato
- identify the event types identificare tipi di eventi
- construct a set of algorithms that define state changes for each event type algoritmi che, in base all'evento, portano al cambiamento dello stato.

Prof. Vittoria de Nitto Personè

8

8

E' un errore grave generare tutti gli arrivi insieme per poi incastrarli con gli altri eventi, poichè l'ordinamento è difficoltoso e artificioso. Si procede sempre passo passo.

## Next-Event Simulation

### Algorithm 1

1. **Initialize** - set simulation clock and first time of occurrence for each event type
2. **Process current event** - scan event list to determine most imminent event; advance simulation clock; update state
3. **Schedule new events** - new events (if any) are placed in the event list
4. **Terminate** - Continue advancing the clock and handling events until termination condition is satisfied

**Note that the simulation clock runs asynchronously; inactive periods are ignored**

Tale orologio è asincrono, salta da 'evento' a 'evento', se ho t=1 evento1, e t=5 evento2, salto da t=1 a t=5, non scansiono t=2,3,4.

scansiono e processo il prossimo evento in lista, eseguendo l'algoritmo associato a tale evento

Quando consumo un evento di un certo tipo, genero il prossimo di quel tipo, altrimenti il processo si chiude!  
Può capitare che e0 generi e2, ma arriva prima e1 e quindi schedo lui

definisco first time PER OGNI evento ove possibile, altrimenti lo setto ad 'impossibile' (es: evento completamente a t=0 lo è) in un server singolo, partendo da sistema vuoto.

termino se raggiungo una condizione di terminazione, altrimenti ritorno allo step 2.

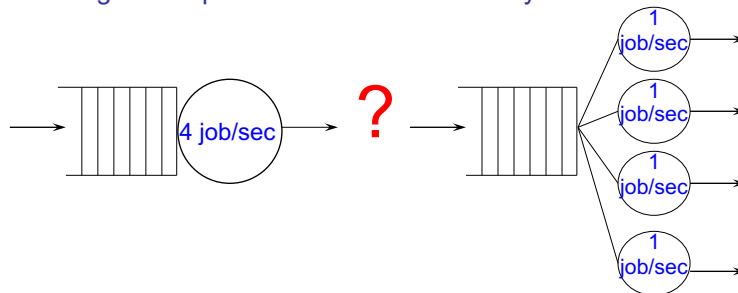
Prof. Vittoria de Nitto Personè

9

9

## Model extensions

Design Example 4: One machine or many?



## A Multi-Server Service Node

Prof. Vittoria de Nitto Personè

10

10

sto cercando di definire cosa includere nella definizione dello stato.

## Conceptual model: MSQ

### Definition 1

A multi-server service node consists of

- A single queue (if any)
- Two or more servers operating in parallel

- **Conceptual model:**  
abstract collection of  
variables and how they  
evolve over time

At any instant in time, poichè siamo al concettuale, ancora non parliamo di stati!

- Each server is either *busy* or *idle*
- The queue is either *empty* or *not empty*
- If one or more servers is idle, the queue must be empty
- If the queue is not empty, all servers must be busy

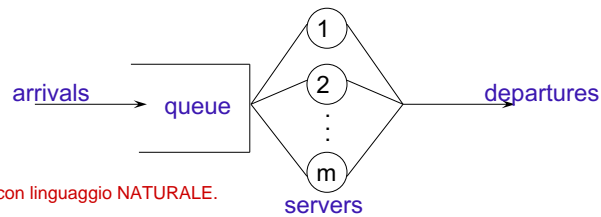
come fosse  
un controllo di  
consistenza

Prof. Vittoria de Nitto Personè

11

11

## Conceptual model: MSQ



qui esplicitiamo ancora con linguaggio NATURALE.

When a job arrives:

- If all servers are busy, the job enters the queue
- Else an idle server is selected and the job enters service

When a job departs:

- If the queue is empty, the server becomes idle
- Else a job is removed from the queue, served by server

*Servers process jobs independently* altrimenti sarebbe un altro modello.

Prof. Vittoria de Nitto Personè

12

12

## Conceptual model: Server Selection Rule

### Definition 2

criterio di scelta del server libero da occupare da parte del job.

The algorithm used to select an idle server is called the *server selection rule*

Common selection rules:

- *random*: at random from the idle servers
- *in order*: lowest-numbered idle server
- *cyclic*: first available, starting after last selected (circular search may be required)
- *equity*: use longest-idle or lowest-utilized
- *priority*: choose the “best” idle server (modeler specifies how to determine “best”)

Se i server sono diversi tra loro, le diverse politiche possono influire sulle prestazioni.