

Modellazione ad Eventi Discreti

Michela Meo

Maurizio M. Munafò

Michela.Meo@polito.it - Maurizio.Munafò@polito.it

Copyright

- Quest'opera è protetta dalla licenza Creative Commons NoDerivs-NonCommercial. Per vedere una copia di questa licenza, consultare: <http://creativecommons.org/licenses/nd-nc/1.0/> oppure inviare una lettera a: Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.
- This work is licensed under the Creative Commons NoDerivs-NonCommercial License. To view a copy of this license, visit: <http://creativecommons.org/licenses/nd-nc/1.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Simulazione ad eventi discreti

- Definizioni (richiamo)
 - *Stato* del sistema ne descrive il comportamento in ogni istante
 - *Entità*: oggetti che operano nel sistema
 - *Evento*: l'occorrenza di qualcosa che determina dei cambiamenti dello stato
 - *Attività*: periodi di tempo per svolgere determinate azioni
 - *Processo*: una sequenza di eventi, ordinata nel tempo, che coinvolgono la medesima entità

Simulazione ad eventi discreti

- Vogliamo modellare tramite un **simulatore ad eventi discreti**
 - Gli eventi sono gli istanti in cui avvengono i cambiamenti di stato del sistema
 - Siamo interessati alla rappresentazione del sistema in questi istanti
 - Il tempo avanza in maniera discreta di evento in evento
 - L'evoluzione del sistema dipende dalla natura degli intervalli tra gli eventi, che possono essere di durata casuale o deterministica

Simulazione ad eventi discreti

- Siamo interessati al sistema nel momento in cui cambia il suo stato, quindi non c'è bisogno di modellare il sistema nei suoi momenti di inattività
- L'approccio scelto è di tipo **next event**, ovvero il tempo avanza di volta in volta in maniera discreta all'istante in cui si verificherà il prossimo cambiamento di stato (evento)

Simulazione ad eventi discreti

- Esistono diverse alternative per la modellazione ad eventi discreti
 - focalizzarsi sulle operazioni da effettuare quando accade ogni singolo evento -> *scheduling degli eventi*
 - considerare le operazioni di ogni entità mentre è presente nel sistema, specialmente in relazione alle altre entità -> *interazione tra processi*

Scheduling degli eventi

- *Schedule*
 - Programmare nel futuro l'esecuzione di un'operazione
- Lo *scheduling degli eventi* prevede quindi che si determini in quale istante di tempo un certo evento dovrà accadere e se ne pianifichi l'esecuzione
- È l'approccio più usato
 - Supportato da quasi tutti i linguaggi di simulazione
 - Molto semplice da codificare

Scheduling degli eventi

- Dobbiamo identificare o definire
 - le entità costituenti il sistema
 - i risultati che desideriamo ottenere dalla simulazione
 - le variabili di stato
 - gli eventi, ovvero ciò che provoca cambiamenti nelle variabili di stato definite

Scheduling degli eventi

- Per ogni evento, dobbiamo definire la sequenza di operazioni da effettuare nel momento in cui l'evento accade
- Queste operazioni sono condizionate dallo stato
 - dell'entità che ha generato l'evento
 - di altre entità del sistema

Scheduling degli eventi

- Operazione speciale e fondamentale effettuata in coincidenza di un evento è lo *scheduling di nuovi eventi*
- Questo significa che, a partire da pochi eventi pianificati all'inizio della simulazione, questa si evolve naturalmente in ragione del rapporto causa/effetto tra gli eventi

Scheduling degli eventi

- Per comprendere a fondo questo approccio, che sarà alla base di quasi tutti i modelli che andremo a studiare, analizziamo passo-passo le fasi di costruzione del modello simulativo per un elemento chiave delle reti di telecomunicazioni: **la coda**

Modello di un sistema a coda

- Consideriamo un sistema a coda
- In tale sistema gli utenti (clienti) accedono in maniera ordinata ad un servizio, fornito da uno o più agenti (servitori) secondo un criterio di ordine prestabilito
- Si tratta del (sotto)sistema più comune che possiamo incontrare nello studio di una rete di telecomunicazione, per il quale esiste una ampia casistica di modellazione analitica

Modello di un sistema a coda

- Quali sono gli elementi di un tale sistema?
 - i clienti che richiedono il servizio
 - i servitori che lo erogano
- A cosa siamo interessati?
 - al tempo medio necessario per ottenere il servizio (tempo di attesa)
 - al numero medio di clienti in attesa di servizio
 - se lo spazio per i clienti in attesa è limitato, la probabilità che un cliente non possa ricevere servizio

Modello di un sistema a coda

- Quali sono le variabili di stato del nostro sistema?
 - il numero di clienti nel sistema
 - lo stato (libero o occupato) del servitore
- Quali sono gli eventi del nostro sistema?
 - l'arrivo di un cliente
 - la fine del servizio per un cliente

Modello di un sistema a coda

- Cominciamo con una versione schematica del nostro modello, per poi arricchirla man mano che identifichiamo i dettagli necessari

■ Arrivo

- Se trovo il servitore libero, lo occupo
- Altrimenti, mi metto in coda

■ Fine Servizio

- Se c'è qualcuno ancora in coda, lo servo
- Altrimenti, divento libero

Modello di un sistema a coda

- Questo primo livello di dettaglio è molto rudimentale
- Mancano le relazioni tra gli eventi
 - Gli eventi di **Fine Servizio** sono causati dalla presenza di clienti e dal fatto che questi vengano effettivamente serviti
- Occorre esplicitare queste relazioni

Modello di un sistema a coda

■ Seconda versione

■ Arrivo

- Se il servitore non è occupato
 - Occupo il servitore
 - Schedulo la fine del servizio
- altrimenti
 - Aspetto in coda

Modello di un sistema a coda

■ Seconda versione

■ Fine Servizio

- Se c'è qualcuno in attesa
 - Scelgo il prossimo in coda
 - Schedulo la fine del suo servizio
- altrimenti
 - Rendo il servitore libero

Modello di un sistema a coda

- Abbiamo evidenziato come la fine di ogni servizio sia pianificata nel momento del suo inizio
- Per il nostro modello simulativo mancano ancora alcuni dettagli essenziali
 - ogni servizio ha una **durata** che, se non fissata, deve essere scelta di volta in volta in maniera adeguata al sistema considerato
 - può essere necessario memorizzare le informazioni dei clienti in attesa in opportune strutture dati

Modello di un sistema a coda

■ Terza versione

■ Arrivo

- Se il servitore non è occupato
 - Occupo il servitore
 - Determino il tempo di servizio T_s
 - Schedulo la fine del servizio per l'istante $T_{curr}+T_s$
- altrimenti
 - Creo un record per il cliente
 - Inserisco il record in coda

Modello di un sistema a coda

■ Terza versione

■ Fine Servizio

- Se c'è qualcuno in attesa
 - Scelgo un cliente da servire
 - Ne estraggo il record dalla coda
 - Determino il suo tempo di servizio T_s
 - Schedulo la fine del servizio per l'istante $T_{curr}+T_s$
- altrimenti
 - Rendo il servitore libero

Modello di un sistema a coda

■ Manca ancora un elemento chiave

Chi fa arrivare i clienti?

- L'arrivo dei clienti è di solito caratterizzato da un processo casuale, ovvero la distribuzione dei tempi di interarrivo
- La soluzione più comune per gestire gli arrivi dei clienti in un sistema è l'*auto-sostentamento*
In coincidenza di ogni arrivo si schedula il successivo
- Un altro punto da sviluppare è la corretta e coerente gestione dei record per i clienti

Modello di un sistema a coda

■ Arrivo

- Calcolo il tempo di interarrivo T_{ia} per il prossimo cliente
- Schedulo un arrivo per l'istante $T_{curr}+T_{ia}$
- Creo un record per il cliente
- Inserisco il record in coda
- Se il servitore non è occupato
 - Occupo il servitore
 - Determino il tempo di servizio T_s
 - Schedulo la fine del servizio per l'istante $T_{curr}+T_s$

Modello di un sistema a coda

■ Fine Servizio

- Estraggo dalla coda il record del cliente che ha finito il servizio
- Distruggo il record
- Se c'è qualcuno in attesa
 - Scelgo un cliente da servire
 - Determino il suo tempo di servizio T_s
 - Schedulo la fine del servizio per l'istante $T_{curr}+T_s$
- altrimenti
 - Rendo il servitore libero

Modello di un sistema a coda

- Il modello può essere ulteriormente dettagliato, evidenziando
 - la politica di gestione della coda
 - la gestione puntuale dei record per i clienti
 - le operazioni necessarie per effettuare le misure desiderate
- Il problema delle misure è molto critico
 - Dove e come vanno realizzate?

Interazione tra processi

- In questo approccio ci si focalizza sulle operazioni svolte da ogni entità nella sua esistenza all'interno del sistema
- La sequenza di eventi e attività che coinvolgono ogni entità costituisce un *processo*
- Il comportamento di un'entità dipende dalle operazioni e dallo stato delle altre entità del sistema
- Dobbiamo evidenziare nel modello l'*interazione* tra le diverse entità e, di conseguenza, tra i processi che le rappresentano

Interazione tra processi

- Il processo rappresenta le operazioni dell'entità con la sua evoluzione nel tempo
- Gli eventi rappresentano ora *sospensioni* e *riattivazioni* del flusso di esecuzione dell'entità
- Due tipi di eventi:
 - *incondizionati* -> il tempo in cui l'evento deve accadere è definito
 - *condizionati* -> l'istante di attivazione dell'evento dipende dal verificarsi di condizioni esterne

Interazione tra processi

- In coincidenza delle sospensioni/riattivazioni, il flusso di controllo delle operazioni passa da un processo all'altro, causando l'evoluzione del sistema
- Il tempo si evolve ancora di evento in evento ma oltre all'insieme degli eventi incondizionati, esiste ora un insieme degli eventi condizionati
- Al termine dell'esecuzione di ogni evento incondizionato
 - si controlla ogni evento condizionato e si verifica se può essere eseguito
 - quando non c'è più nessun evento condizionato eseguibile, si passa al prossimo evento incondizionato, facendo avanzare il tempo

Interazione tra processi

- Appaiono due operazioni particolari
 - "Aspetta per ΔT unità di tempo..." -> il flusso delle operazioni viene sospeso fino all'esecuzione di un evento incondizionato nell'istante $T+\Delta T$
 - "Aspetta finché..." -> il flusso delle operazioni viene sospeso indefinitamente finché la condizione associata non risulta soddisfatta

Interazione tra processi

- Possiamo comprendere questo approccio modellando il sistema a coda
- Le entità del sistema abbiamo visto essere
 - i clienti
 - il servitore
- Modelliamo quindi
 - un processo per le operazioni svolte da ogni singolo cliente
 - un processo per le operazioni svolte dal servitore

Interazione tra processi

■ Cliente

- Calcolo il tempo di interarrivo T_{ia} per il prossimo cliente
- Schedulo un arrivo per l'istante $T_{curr} + T_{ia}$
- Creo un record per il cliente
- Inserisco il record in coda
- **Aspetto finché** non è finito il servizio
- Distruggo il record

Interazione tra processi

■ Servitore

- Se la coda è vuota
 - **Aspetto finché** non arriva un cliente
- Scelgo un cliente da servire
- Estraggo dalla coda il record corrispondente
- Determino il tempo di servizio T_s
- **Aspetto per T_s unità di tempo**, fino alla fine del servizio (all'istante $T_{curr} + T_s$)

Interazione tra processi

- Il modello del servitore è ciclico e non termina mai: occorre programmarne l'attivazione una sola volta, all'inizio della simulazione
- Possono essere contemporaneamente presenti numerosi processi relativi ai clienti, uno per ogni cliente in attesa, mentre ci sarà sempre e solo un processo di servitore
- I punti di sospensione/riattivazione sono evidenziati in blu e rappresentano gli istanti in cui il controllo passa da un processo all'altro

Interazione tra processi

- L'interazione tra processi è un approccio modellistico molto potente
 - La descrizione del comportamento dell'entità è concentrata in un unico blocco, invece di essere frammentata tra diversi eventi
- La semplicità rappresentativa del modello però nasconde la complessità implementativa legata a
 - eventi condizionati/incondizionati
 - punti di riattivazione del flusso di elaborazione