Modellazione discreta, esempi con GS DSLibs

Simulazione & Logistica, I modulo
Esercitazione n. 6

Corso di Laurea in Informatica Applicata
Università di Pisa, sede di La Spezia
A.a. 2008/09, I semestre

Contenuti

- L'impiegato tormentato, varie variazioni e altro
- Cambiamenti funzionali
- Cambiamenti di comportamento
- Eventi ritirati
- Generatori di oggetti
- Eventi a tempo zero

3+1 variazioni al caso di studio

- HClerk0, il caso di studio originale
  - I clienti arrivano di persona, oppure telefonano
  - L'impiegato serve i clienti allo sportello o al telefono
  - I clienti che arrivano allo sportello aspettano (in ordine)
  - Il centralino ordina le telefonate in attesa
  - I clienti allo sportello hanno la priorità
  - I tempi di servizio dipendono dai clienti
- Variazioni
  - Obiettivi di simulazione e variazioni funzionali
  - Clienti impazienti (classico e con GS DSLibs)
  - Generatori di clienti

HClerk1, prima variazione

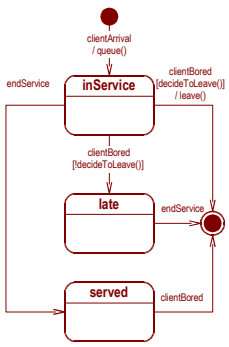
- Politica di servizio parametrica
  - Priorità ai clienti allo sportello finché la coda al centralino non supera la coda allo sportello di un Δ dato
  - $\Delta = \infty$, la politica coincide con il caso originale
  - $\Delta = 0$, si serve la coda più lunga (a parità, lo sportello)
- Obiettivi di simulazione, rispetto a politiche diverse
  - Tenere traccia dei tempi di servizio totali
  - Tenere traccia del tempo in cui l'impiegato "non lavora"
- Cambiamenti funzionali
- Cosa cambia nel modello? E nel codice?

HClerk2, seconda variazione

- Comportamento diverso: clienti impazienti
  - Disponibilità limitata ad aspettare
  - Un attributo in più che caratterizza tutti i clienti
  - Esaurita la pazianza se ne vanno o riattaccano
- Clienti in coda allo sportello
  - Conoscono la loro posizione in coda
  - Valutano se andarsene in base alla posizione raggiunta
  - Un altro attributo, caratterizza solo i clienti allo sportello
- Obiettivi di simulazione (aggiuntivi)
  - Tenere traccia dei clienti che se ne vanno o riattaccano
  - Tenere traccia dei clienti serviti in ritardo
- Cosa cambia nel modello? E nel codice?

HClerk2a, classico

- Un evento in più
  - Per avvisare che la pazienza è arrivata al limite
- Da gestire
  - Anche quando non serve
  - Occorre uno stato esplicito nel caso in cui il cliente sia già stato felicemente servito

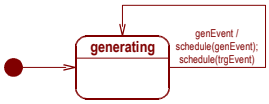


HClerk2b, con GS DSLibs

- Se mi sveglio prima, spengo la sveglia
- Metodo withdraw()
  - Disabilita un evento in coda
  - I clienti serviti disabilitano i loro eventi di fine attesa
- È corretto modificare il futuro?
- Condizioni imposte:
  - L'evento è riferito tramite il suo identificatore unico, e solo conoscendolo può essere disabilitato
  - Un evento può essere disabilitato solo dall'istanza di entità attiva che lo ha programmato (warning se non)
- Si può modificare il proprio futuro, che è già noto

HClerk3, terza variazione

- Generatori di oggetti
  - Evitare di caricare l'agenda prima di lanciare la simulazione
  - Suggerimento che viene dalla modellazione per attività
- Entità dedicate
  - Molto semplici
  - Rigenerano gli eventi
  - Informazione confinata



- Alternative per l'implementazione
  - Non aggiungere eventi, modificare la doAction() per includere il generatore fra i gestori
  - Aggiungere eventi dedicati di generazione

Eventi a tempo 0

- Modellazione classica discreta
  - Evento: un dato istante
  - Azione associata: tutto quello che succede a quell'istante
- Programmazione di eventi a tempo 0
  - Estranea alla logica del modello
  - Non supportata dalle architetture
- A volte utile
  - Generatori, eventi contemporanei vanno previsti
  - Segnali, temporalmente successivi, ma con Δ trascurabile
  - Separazione della logica delle entità
  - Controllori e punti di sincronizzazione