

Sistemi IV

(Corso di Informatica, 5 anno)

Linguaggi, Strumenti e modelli di Simulazione

Luciano Bononi

bononi@cs.unibo.it

<http://www.cs.unibo.it/~bononi>

Ricevimento: Lun-Ven 9-18 :-)

presso ufficio dottorandi, Mura Anteo Zamboni 7, Bologna

Simulazione: Qnap2

- QNAP2
 - sistema completo per gestire, descrivere e risolvere modelli basati su reti di code
 - comprende un linguaggio di specifica dei modelli e un set di moduli di risoluzione del modello (solvers) basati su tecniche diverse

Simulazione: Qnap2

- QNAP2: il linguaggio
 - configurazione della rete di code
 - set di stazioni (= singola coda + server(s))
 - customers distributed over classes
 - le classi caratterizzano diversi comportamenti dei customers oppure diverse elaborazioni a loro carico
 - routing rules (non dinamiche! Compile time...)
 - elaborazione compiuta in ogni stazione
 - una semplice attesa caratterizzata da media e distr.
 - ...oppure un complesso algoritmo+sincronizzazione
 - controllo della risoluzione del modello
 - inizializzazione, aggiornamento dei parametri, e sequenzializzazione delle fasi e metodi di risoluzione e analisi dei risultati

Simulazione: Qnap2

- QNAP2: il linguaggio
 - esistono due livelli del linguaggio
 - un linguaggio di controllo
 - un linguaggio algoritmico (derivato da Simula e Pascal) per descrivere nel dettaglio gli statements di controllo
 - ...ed esistono tre tipi di statement di controllo
 - stand-alone commands: es. /TERMINALS/, /RESTART/, /END/
 - parametrized commands: es. /DECLARE/, /STATION/, /CONTROL/ di solito assegnano valori ai parametri e sono “blocchi algoritmici”
 - comando /EXEC/: blocco algoritmico

Simulazione: Qnap2

- QNAP2: il linguaggio
 - un programma QNAP2 è una sequenza ordinata di comandi (algoritmici e di controllo) che permettono di definire
 - inizializzazione e definizione completa del modello
 - descrizione del modello
 - controllo dell'analisi
 - risoluzione del modello

Simulazione: Qnap2

- QNAP2: i meccanismi di risoluzione
 - i solver sono di quattro tipi
 - discrete event simulation:
 - OK per qualsiasi modello (generale)
 - costoso in termini di calcolo e infrastruttura
 - confidence intervals e transient state analysis
 - Markov chain analyser
 - risultati esatti, ma solo steady state
 - solver analitici esatti
 - basati su teoremi, oppure MVA oppure convoluzione
 - solver analitici approssimati
 - metodi iterativi, metodi a diffusione, approcci euristici

Simulazione: Qnap2

- QNAP2: il linguaggio algoritmico
 - parole riservate
 - notazione numerica: N.B. il reale .46 è scorretto.
 - Stringa : “testo della stringa”
 - commento : & questo è commento
 - separatore : tutto tranne TAB
 - il carattere slash / in colonna uno è **COMANDO**

Simulazione: Qnap2

- QNAP2: il linguaggio algoritmico
 - Tipi
 - scalari (non strutturati)
 - integer, real, boolean, string
 - array (strutturati)
 - ad n dimensioni con oggetti dello stesso tipo
 - es. integer numero(20)
 - oggetti (strutturati)
 - tipi predefiniti (ed estendibili) sono: QUEUE, CUSTOMER, CLASS, EXCEPTION, FLAG, TIMER, FILE
 - possono essere definiti oggetti complessi (record e liste)
 - riferimenti (puntatori): ref

Simulazione: Qnap2

- QNAP2: il linguaggio algoritmico
 - statement di dichiarazione /DECLARE/
 - vale per tipi, variabili, procedure e funzioni
 - espressioni
 - valgono le solite regole sugli operatori classici
 - NOT, **, ::, IS, IN
 - N.B. operatori aritmetici su interi danno interi, reali o.w.
 - TRUE > FALSE
 - $a > b$ and $c > d$ è errato : $(a > b)$ and $(c > d)$ è corretto
 - costanti:
 - numeriche
 - NIL(puntatore nullo),
 - TRUE, FALSE

Simulazione: Qnap2

- QNAP2: il linguaggio algoritmico
 - chiamate di funzione (con e senza parametri)
 - function id [sublist...]
 - liste e sottoliste
 - operatori ALL, WITH, REPEAT, STEP UNTIL
 - statements (semplici e composti begin...end)
 - goto, if..then..else, for...do, while..do, with..do
 - I/O
 - input: GET, GETLN
 - output: PRINT fixed length+\n, WRITE, WRITELN
 - gestione file: FILEASSIGN, OPEN, CLOSE

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - definisce la struttura del “programma”
 - /declare/ dichiarazione identificatori
 - /station/ descrizione caratteristiche della stazione di lavoro
 - /control/ specifica parametri di controllo di QNAP2
 - /terminal/ esecuzione interattiva di QNAP2
 - /exec/ blocchi di istruzioni algoritmiche
 - /reboot/blocco di reboot dopo un restore
 - /restart/ introduce un nuovo modello
 - /END/ (è solo il terminatore del sorgente)

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - valutazione dei parametri
 - compile time (termina quando incontro /exec/)
 - execution time
 - initiation time: inizia quando chiamata da exec
 - qui valuta il lato destro dei parametri delle TRANSIT dei comandi /STATION/
 - solution time: inizia quando exec chiama il solver
 - qui valuta le espressioni dei parametri dei servizi (SERVICE) in /STATION/

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - /DECLARE/
 - introduce lista di dichiarazioni di identificatori
 - possono essere multiple
 - valutate a compile time
 - /EXEC/
 - introduce ed esegue comandi algoritmici
 - inizializzazione di variabili, chiamata a procedure, stampa risultati
 - esecuzione: execution time, subito dopo la compilazione dell'exec e subito prima della compilazione del resto del programma (model alteration)

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - /TERMINAL/
 - lavoro interattivo su QNAP2
 - /RESTART/
 - ripristina lo stato iniziale dell'esecuzione del modello
 - /REBOOT/
 - re-inizializza alcune var e riparte la simulazione dal punto in cui era stata interrotta

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - /STATION/
 - describe una o più stazioni del modello
 - una stazione describe UNA CODA e i suoi server(s)
 - può essere usato per definire o alterare caratteristiche delle stazioni del modello
 - parametri:
 - NAME, TYPE, SCHED, INIT, SERVICE, TRANSIT, SPLIT, MATCH, FISSION, FUSION, RATE, CAPACITY, REJECT, PRIOR, QUANTUM, COPY
 - vediamo i più significativi (pag. 80, cap. 3)

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - NAME
 - deve essere il primo parametro
 - uso * per identificare una template queue
 - usato per definire oggetti come sotto-tipi di QUEUE
 - descrizione di code definite come attributi di oggetti
 - valutate a compile time
 - TYPE: default=single se esiste attrib.SERVICE, altrimenti è solo una coda senza server
 - server: SINGLE, MULTIPLE(expr), INFINITE
 - resource: come sopra per specificare il numero di ris.
 - semaphore: SINGLE (s=1), MULTIPLE(expr) (s=n)
 - source: definisce una station sorgente
 - multiple(n): definisce una station di tipo server

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - SCHED
 - FIFO, LIFO, FILO, PRIOR(CPRIOR), PREEMPT(p), EXCLUDE(mutex) per evitare servizi concorrenti da parte di server multipli su customer appartenenti allo stesso concurrency set, FEFS. QUANTUM, PS, RESEQUENCE
 - PRIOR(exp): exp valutata a compile time!
 - determina il livello di priorità del customer che entra nella station (in classe k)
 - priorità minima = 0

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - QUANTUM: valutato ad init. time
 - definisce il quantum di servizio della stazione assegnato ciclicamente ad ogni classe (ovviamente se SCHED=Quantum)
 - INIT: compile time
 - definisce il numero di customer IN OGNI CLASSE della coda della stazione (attenzione!)
 - SERVICE
 - determina il servizio al quale il customer è soggetto, tranne che per station = risorse e semafori
 - include work demand + operazioni di manipolazione
 - la manipolazione può definire P o V, TRANSIT, PRIOR...che controllano l'attività dei server della stazione

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - TRANSIT
 - descrive le regole di routing dei customer al momento del completamento del servizio da parte della station
 - è uno switch probabilistico (con prob. valutate a init time!)
 - lista di triple: (q, [c,] r) dove q=lista di code, c=lista opzionale di classi, r=lista di prob. relative o assolute
 - es. TRANSIT = A, 0.5, B, 0.3, C (prob. assolute)
 - es. TRANSIT = A, 10, B, 6, C, 4 (prob. relative)
 - es. TRANSIT(classek) = A, PA, B, Y, PB, C, Z
 - » dalla classe k della station un customer migra verso la station A con prob PA, verso la classe y della station B con prob PB, e verso la classe z della station C con prob 1-PA-PB
 - es. TRANSIT = A, 0.2, OUT

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - SPLIT
 - come TRANSIT, ma esegue uno split del customer
 - sono triple (q, c, n) dove q=coda destinazione, c=classe destinazione, n=numero di customer generati dallo split
 - anche qui le regole di routing sono switch probabilistici
 - MATCH
 - definisce il join di customer precedentemente splittati
 - vedi manuale per dettagli (pag. 94)

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - FISSION
 - come TRANSIT ma spezza il customer in più istanze senza tenere memoria dell'origine
 - FUSION
 - come MATCH ma non tiene conto della origine dei customer “fusi”
 - RATE
 - definisce alterazioni alla velocità di servizio di uno o più server della station
 - CAPACITY
 - massimo numero di customer presenti nella station

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - REJECT
 - specifica l'elaborazione che devono subire i customer respinti dalla station, a causa del limite di CAPACITY
 - COPY
 - esegue copia (compile time) di una STATION predefinita

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - questi erano i parametri del comando /STATION/
 - vediamo ora il comando /**CONTROL**/
 - modifica il controllo dei parametri di QNAP2
 - parametri di I/O
 - OPTION, UNIT
 - parametri di risoluzione
 - CLASS, MARGINAL, ENTRY, EXIT, CONVERGENCE
 - controllo della simulazione
 - TSTART, TMAX, PERIOD, RANDOM, TRACE, ACCURACY, ESTIMATION, STATISTICS, TEST, CORRELATION
 - altro: NMAX, ALIAS

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - OPTION
 - controllo stampa dei risultati (procedura OUTPUT), trace degli eventi, trace del debugger, attivazione test durante la simulazione, stampa messaggi di warning, attivazione della parallelizzazione delle repliche
 - UNIT
 - controlla i file di I/O
 - OUTPUT: stampa ciò che è definito in OPTION
 - PRINT: stampa (PRINT, WRITE, WRITELN)
 - INPUT: è il file del program source
 - GET: definisce il file di input (default) per GET, GETLN
 - LIBR: libreria di default per SAVE, RESTORE
 - TRACE: standard trace output file

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - CLASS
 - SETSTAT:CLASS invoca una procedura per il calcolo delle statistiche sulla classe definita
 - MARGINAL
 - SETSTAT:MARGINAL invoca una procedura per il calcolo delle statistiche marginali
 - CONVERGENCE
 - modifica parametri per i solver analitici

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - TSTART
 - è il valore del tempo (simulato) al quale devono iniziare le stime dei parametri da valutare con la simulazione. Ciò permette di eliminare l'influsso delle fasi transienti dalle stime ottenute.
 - Default=0
 - se il valore è EXPR viene valutato a init. Time
 - TMAX
 - durata massima di un RUN di simulazione
 - default=1000, valutato a init. Time
 - la procedura SETMAX permette il settaggio runtime

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - RANDOM
 - inizializza il generatore pseudocasuale
 - default=413
 - il generatore pseudocasuale non viene re-inizializzato se faccio diversi RUN all'interno di un singolo /EXEC/ command (uso un singolo stream)
 - /EXEC/ SIMUL
 - /EXEC/ SIMUL usano lo stesso stream!
 - /CONTROL/ RANDOM=337 (init un nuovo stream)
 - /EXEC/ SIMUL

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - ACCURACY
 - specifica le stazioni e le classi per le quali si richiedono le stime degli intervalli di confidenza
 - es. ACCURACY = ALL QUEUE, ALL CLASS
 - es. ACCURACY = NIL
 - ESTIMATION
 - definisce il metodo per la stima degli intervalli di confidenza tra Replication, Regeneration, Spectral

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - STATISTICS
 - quando il metodo di stima è Regeneration
 - PARTIAL: la chiamata a OUTPUT durante un RUN stampa i risultati parziali delle stime durante la rigenerazione attuale (delta t tra ultime 2 chiamate della procedura SAMPLE)
 - GLOBAL: la chiamata a OUTPUT durante un RUN stampa le statistiche ottenute dal TSTART fino all'ultimo regeneration point (ultima chiamata di SAMPLE).
 - Alla fine della simulazione entrambi i parametri danno gli stessi risultati (statistiche su tutto il RUN)
 - CORRELATION
 - stima di correlazione delle misure effettuate

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - PERIOD
 - definisce il periodo di attivazione della TEST sequence (vedi TEST parameter)
 - period=0 significa che TEST viene eseguita dopo ogni evento
 - TEST
 - definisce una sequenza di statement che sono eseguiti alle scadenze definite da PERIOD
 - es. chiama SAMPLE per ridefinire le rigenerazioni
 - es, chiama OUTPUT per stampare dati temporanei (transiente)
 - es. chiama STOP per interrompere il RUN prima di TMAX

Simulazione: Qnap2

- QNAP2: il linguaggio di controllo
 - TRACE
 - definisce le opzioni per il trace file
 - NMAX
 - definisce il numero massimo di classi possibili nel modello (area di RAM allocata)
 - default=20
 - ENTRY
 - definisce una sequenza di comandi da eseguire come entry point del primo run di simulazione
 - EXIT
 - definisce una sequenza di comandi da eseguire al termine dell'ultimo run di simulazione

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - modeling su reti di code, basato su QNAP2
 - Queue organization
 - Station
 - Service
 - Server allocation
 - Customer priority, creation, join, transition, split, match
 - semaphores
 - resources
 - flags
 - exceptions
 - timers

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Queue organization
 - Queue structure: lista bilinkata
 - attributi .PREVIOUS (verso l'inizio) , .NEXT (verso fine)
 - attributo CODA.FIRST punta al primo customer (NIL se vuota, customer sotto servizio se single server busy)
 - attributo CODA.LAST punta all'ultimo customer
 - coda vuota:
FIRST=LAST=FIRST.PREVIOUS=LAST.NEXT=NIL

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Queue organization
 - Queue discipline
 - FIFO: --> (last)(first) [server(s)]
 - LIFO: (last)...(first)<-- [server(s)]
 - FIFO, PRIOR: ogni customer ha attributo CPRIOR che può essere modificato dalle procedure TRANSIT, BEFCUST, AFTCUST, PRIOR, o dal parametro PRIOR della station. I customer a priorità alta sono vicino alla testa della coda (vicino al server). A parità di PRIOR sono FIFO.
 - LIFO, PRIOR: come sopra. Ma a parità di PRIOR sono LIFO.
 - RESEQUENCE: i customer sono ordinati nell'ordine esatto col quale sono partiti dalla reference queue.

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Queue organization
 - Queue capacity
 - parametro CODA.CAPACITY[(class)]
 - default= infinito (-1)
 - Queue initialization
 - STATION.INIT parameter definisce il numero di customer iniziale in ogni CLASSE della station

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Station
 - single server station
 - multiple server station
 - infinite server station (delay node)
 - source station (SERVICE definisce i tempi di interarrivo, e non di servizio)

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Service
 - work demand
 - soggetta a media, distribuzione, ecc.
 - la procedura FREE libera immediatamente un customer sotto servizio (senza possibilità di RESUME)
 - object manipulation operations
 - durante un servizio il customer può manipolare oggetti
 - » customer, flag, semafori, risorse
 - per compiere tali operazioni, il customer deve avere un server assegnato. Tali operazioni non consumano tempo di simulazione.
 - Se un customer sotto servizio richiede una risorsa (non disponibile) il server e il customer diventano bloccati. Quando la risorsa è disponibile il servizio riprende (RESUME) da dove era rimasto.

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Service
 - service completion
 - il customer che completa il servizio esegue un trasferimento specificato dal parametro TRANSIT della STATION (statico). Transiti dinamici possono essere specificati mediante procedure TRANSIT, MOVE, BEFCUST, AFTCUST
 - service suspension
 - preemption su server o su risorse in uso
 - blocking behavior (richiesta di risorse non disponibili)
 - server allocation (continua pag 135)

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - server allocation
 - specificata dal parametro SCHED
 - avviene quando almeno un customer si trova in coda
 - il customer detiene il server (e risulta servito da esso) anche quando il servizio risulta bloccato per qualche motivo (N.B. un customer in attesa di risorse e in coda per un server, viene allocato sul server e lo impegna, anche se il customer è di fatto bloccato in attesa della risorsa e il server non gli fornisce nessun servizio attivo!)
 - il customer che non viene servito si dice “waiting”

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - server allocation
 - SCHED: fifo, lifo, Quantum, PS...
 - SCHED: resequencing station
 - i customer sono serviti dal server nello stesso ordine con il quale sono partiti dalla station di riferimento. I customer che arrivano prima in coda semplicemente aspettano finchè i loro predecessori non hanno ricevuto il servizio.
Attenzione, perché il server può rimanere bloccato all'infinito pur avendo customer in coda, se uno dei predecessori rimane bloccato o distrutto.

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Customer priority level
 - parametro CUSTOMER.CPRIOR ≥ 0
 - N.B. se definisco $PRIOR(X) = 1$ come parametro di /STATION/, ottengo che tutti i customer che entrano in classe X vedono assegnarsi CPRIOR=1, altrimenti manterrebbero il valore inalterato
 - N.B. il CPRIOR può essere cancellato o forzato durante una transition attraverso le seguenti procedure (TRANSIT, P, BEFCUST, AFTCUST)
 - `PRIOR([customer],integer)`
 - `TRANSIT(where, CPRIOR,...)` vai a where con CPRIOR

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Customer Creation
 - NEW(CUSTOMER) crea un customer “apolide” che deve essere TRANSITato verso una coda (CQUEUE attribute=NIL)
 - JOIN e JOINC([cust],n | cust_list): sono procedure che determinano l’attesa del customer cust finchè n suoi figli (oppure i figli citati nella cust_list) non sono distrutti (OUT, SPLIT, FISSION)
 - JOIN agisce sul current CUSTOMER, mentre JOINC agisce sul customer espressamente citato

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Customer Forced Transition
 - procedura TRANSIT([cust],queue[,class][,prior])
 - forza la transition del customer *[cust]* nella classe *class* della coda *queue*, assumendo CPRIOR=*prior*.
 - N.B.
 - » se il customer è sotto servizio, la TRANSIT invocata su di esso interrompe il servizio!...senza RESUME!
 - » Se queue=OUT il customer viene distrutto
 - » Un customer bloccato o che detiene risorse non può essere distrutto
 - » gli statement che seguono TRANSIT per il customer corrente non verranno MAI eseguiti!

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Customer Forced Transition
 - procedura MOVE(queue1,queue2[,class][,prior])
 - forza la transition del primo customer della coda *queue1* nella station della coda *queue2*, in classe *class* e con CPRIOR=*prior*.
 - Procedura BEFCUST(cust1,cust2[,class][,prior])
 - forza la transition del customer *cust1* nella classe *class* della coda che contiene il customer *cust2*, prima del *cust2*, con CPRIOR=*prior*.
 - Procedura AFTCUST(cust1,cust2[,class][,prior])
 - forza la transition del customer *cust1* nella classe *class* della coda che contiene il customer *cust2*, dopo il *cust2*, con CPRIOR=*prior*.

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Customer Split and Match
 - SPLIT: spezza un singolo cust in tanti custs che possono essere indirizzati su varie classi e code. Rimane un vincolo di associazione al cust padre.
 - MATCH: vari custs generati da SPLIT sono unificati in un singolo cust
 - FISSION: come SPLIT, ma senza associazione al padre.
 - FUSION: come match, ma senza check di paternità comune
 - N.B. sub-type customer generano customer, e le unificazioni avvengono in ingresso alla station! (i customer in attesa attendono “fuori” dalla station)

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Semaphores
 - sono STATION: coda + un contatore
 - un customer può richiedere P durante un servizio, o può forzare altri customer a chiedere P in qls. momento
 - possono essere creati dinamicamente come istanze di oggetti QUEUE o loro sottotipi.
 - /STATION/ type=semaphore, [multiple(integer)]
 - hanno solo attributi PRIOR e SCHED

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Semaphores
 - P ([cust,]queue[,class][,prior]) procedure
 - se un customer esegue una P bloccante, una copia del customer viene inserita in coda al semaforo con classe e prior specificata o ereditata.
 - Per liberare un customer in coda occorre una V sul semaforo, oppure una call della procedura FREE sul customer bloccato
 - N.B. un cust può chiedere P multiple
 - N.B. non ha senso la preemption per i semafori

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Semaphores
 - V (queue) procedure
 - se un customer esegue una P bloccante, una copia del customer viene inserita in coda al semaforo con classe e prior specificata o ereditata.
 - Per liberare un customer in coda occorre una V sul semaforo, oppure una call della procedura FREE sul customer bloccato
 - N.B. un cust può chiedere P multiple
 - N.B. non ha senso la preemption per i semafori
 - N.B. un customer può fare V su un semaforo anche se non ha mai fatto P su di esso (in questo è la differenza rispetto alle resources).

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Semaphores
 - Pmult([cust],[priorlist,]queuelist[,classlist][,int_list])
 - Vmult ([cust,]queuelist[,int_list])
 - si applicano a code di tipo semaforo o risorsa
 - effettuano in maniera atomica P e V multiple (quindi non sono equivalenti a una sequenza di P o di V)!!!

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Resources
 - sono STATION= una coda e uno o più server passivi
 - si eseguono le solite operazioni di P e di V (anche Pmult e Vmult)
 - P([cust],queue[,class][,prior]
 - forza CUST a richiedere una risorsa dalla coda QUEUE
 - un nuovo cust viene creato e messo in coda per la risorsa, nella classe CLASS e con priorità PRIOR (il cust originale è bloccato in attesa della risorsa. Appena la risorsa viene ottenuta dal cust copia, il cust originale viene sbloccato, ma la copia rimane in coda finchè non verrà fatta la V).
 - V([cust],queue)
 - N.B. un cust non può fare V su una risorsa dove non ha fatto P in precedenza (al contrario dei semafori).

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Flag
 - servono a sincronizzare customer paralleli
 - ogni flag ha due stati: set (TRUE) , unset (FALSE)
 - creazione: NEW(flag)....default=unset
 - gestione: procedure SET, UNSET, RESET
 - utilizzo: procedure WAIT, WAITAND, WAITOR
 - es. WAIT([cust,]flag), WAITAND([cust,]flag_list)
 - il customer attende nel punto in cui si trova (eventualmente mantenendo un server) finchè il FLAG diventa TRUE. Un customer in attesa è sbloccato da SET o da una FREE
 - N.B. i customer waiting sono accodati LIFO in una coda di attesa, affinché sia possibile scandire la coda e fare operazioni su di essi. <FLAG>.LIST punta al primo customer della lista, <cust>.link punta al prossimo in lista.

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Exceptions
 - usate per catturare eventi speciali (interrupt, condizioni di errore, ecc.)
 - ogni exception ha una procedura “handler” invocata automaticamente al verificarsi dell’evento.
 - SIMSTART, SIMACCUR, SIMSTOP sono esempi legati alla gestione della simulazione

Simulazione: Qnap2

- QNAP2: il meccanismo di Modeling
 - Timers
 - servono a modellare processi indipendenti dalla QN
 - sono simili a PERIOD, TMAX, TSTART, ma possono essere definiti dall'utente
 - possono essere usati solo in simulazione, e sono associati a un “handler” che viene invocato quando il timer è scaduto
 - N.B. sono timer relativi al tempo simulato, non al tempo reale!!!
 - Esiste il modo di definire priorità sulle chiamate degli handler di timer che scadono contemporaneamente.

Simulazione: Simul in Qnap2

- Procedura SIMUL in Qnap2
 - discrete event simulation del modello
 - riproduce evoluzione temporale del comportamento di stations, customers, variables...
 - sequenza di istanti in cui si verificano eventi a carico del modello
 - monitoring automatico dell'evoluzione al fine di trarre stime delle prestazioni (2 tipi principali)

Simulazione: Simul in Qnap2

- Queue statistics
 - service time, response time, blocked time, utilization rate, number of customers, throughput
- User statistics
 - su variabili definite dall'utente come “watched”
 - le variabili watched sono modificate da codice algoritmico dell'utente e sono tracciate dal simulatore ad ogni loro modifica.

Simulazione: Simul in Qnap2

- Analisi delle fasi transiente e stazionaria
 - il simulatore permette entrambe le analisi
 - Global statistics: usate per steady-state analysis
 - sono calcolate su tutto il run di simulazione, tolti gli intervalli nei quali le statistiche sono disabilitate
 - Assunzione di base per l'analisi dei risultati: il sistema raggiunge lo stato stazionario se la simulazione dura un tempo infinito
 - maggiore lunghezza del run --> maggiore accuratezza statistica della stima (Qnap2 fornisce stima automatica degli intervalli di confidenza)
 - Partial statistics: usate per analisi transienti
 - calcolate su periodi fissati (startup, transizioni tra stati stazionari, sistemi non stazionari...)

Simulazione: Simul in Qnap2

- Controllo della lunghezza della simulazione
 - parametro TMAX: massima durata del run
 - espresso secondo la scala temporale del modello
 - il run può essere interrotto PRIMA di TMAX
 - se si invoca una procedura STOP da parte di un qualsiasi controllo algoritmico della simulazione (N.B. questo è un modo “pulito” di uscire dalla simulazione, in quanto le stime sono calcolate comunque, e tutte le procedure che seguono la procedura SIMUL sono eseguite in seguito. Per interrompere il run e uscire da QNAP si deve usare ABORT.
 - se il livello di accuratezza richiesto per le stime risulta raggiunto

Simulazione: Simul in Qnap2

- Il controllo del livello di accuratezza
 - intervalli di confidenza su stima dei valori medi
 - possono essere calcolati su queue e su user stat.
 - Metodi classici (vedi appunti)
 - può definire quando il run è lungo abbastanza
 - ciò permette di risparmiare tempo di cpu
 - possono essere definite regole di accuratezza, e il run termina automaticamente quando tutte le regole sono soddisfatte

Simulazione: Simul in Qnap2

- La TEST sequence
 - è una sequenza eseguita periodicamente (parametro PERIOD del comando /CONTROL/ durante il run, che permette
 - accesso a variabili globali e test di condizioni
 - stampa di valori run-time (tracce di parametri)
 - stampa di statistiche del modello (OUTPUT proc.)
 - interrompere il run di simulazione (STOP proc.)
 - definire i punti di rigenerazione del sistema (se selezionato il metodo di rigenerazione)
 - TEST non permette
 - creazione e transizioni di customers

Simulazione: Simul in Qnap2

- Stima degli intervalli di confidenza
 - livello di confidenza CL, es. 95%
 - prob. di stima esatta entro i limiti dell'intervallo è pari a 0.95
 - 3 metodi usati:
 - metodo delle prove ripetute (replication)
 - metodo di rigenerazione (regeneration)
 - esiste stato di rigenerazione? (passato non influisce su futuro)
 - metodo spettrale (spectral): (studia correlazione...)
 - deve essere scelto e parametrizzato attraverso il comando /CONTROL/ e relativi parametri.

Simulazione: Simul in Qnap2

- Risultati di SIMUL
 - stime di QUEUE e USER statistics
 - watched variables: mean, min, max, variance
 - **mean service time:**
 - calcolato considerando solo i customers che hanno completato il loro servizio
 - tempo medio durante il quale i customers che hanno completato il servizio hanno occupato il server (o la coda k del server): $S(q [,k]) / N(q [,k])$

Simulazione: Simul in Qnap2

- Risultati di SIMUL (QUEUE stats.)
 - **Mean response time**
 - calcolato considerando solo i customers che hanno completato il loro servizio
 - tempo medio passato sul (coda q + server) dai customers che hanno completato il servizio (anche di classe specificata k): $R(q [,k]) / N(q [,k])$

Simulazione: Simul in Qnap2

- Risultati di SIMUL (QUEUE stats.)
 - **mean number of customers**
 - numero medio di customer che hanno lasciato la stazione e che si trovano ancora nella stazione (non discrimina tra customer in coda o sotto servizio):
 - $(\text{Sum}_{i=1..inf} i * T(q [,k], i) / N(q [,k])) / T_m$

Simulazione: Simul in Qnap2

- Risultati di SIMUL (QUEUE stats.)
 - **mean blocked time**
 - tempo medio durante il quale un customer è bloccato
 - in attesa di risorsa, su un semaforo, su join o su un flag
 - considera solo i customer che hanno completato il servizio
 - N.B.
 - BLOCK e UNBLOCK bloccano il server e non i customers
 - il tempo durante il quale è bloccato il server non viene contato per i suoi customers

Simulazione: Simul in Qnap2

- Risultati di SIMUL (QUEUE stats.)
 - **(mean) utilization rate**
 - tempo medio di occupazione di un server (o risorsa) tra tutti gli M server o risorse della stazione, da parte di almeno un customer durante la sessione
 - $(S'(q [,k]) / T_m) / M$
 - N.B.
 - singolo server = tempo medio di occupato
 - coda semplice (no server) = valore NULL (sempre)
 - infinite server (delay) = sempre ZERO

Simulazione: Simul in Qnap2

- Risultati di SIMUL (USER stats.)
 - TSTART: tempo iniziale di stima delle var. watched
 - in alternativa SETSTAT: ON/OFF var. list
 - GESTAT: genera report dei dati specificati
 - maggiori dettagli sul manuale QNAP2 User's Guide

Simulazione: Simul in Qnap2

- Note sul processo di simulazione
 - ordine di esecuzione dei processi **customer**,
(timer ed exceptions sono attivati da chiamate)
basato su
 - linea temporale, priorità, tempo di pianificazione
 - un customer diviene attivo se
 - viene allocato un server per esso
 - non è bloccato su semaforo, risorsa, flag o join
 - deve transitare tra nodi al tempo attuale

Simulazione: Simul in Qnap2

- Note sul processo di simulazione
 - un customer diviene inattivo se
 - chiama una work demand procedure (anche NULL)
 - fine del quantum
 - entra in stazione dove non ci sono server disponibili
 - esegue P, WAIT, JOIN che lo blocca
 - esegue V, SET, FREE, TRANSIT, MOVE che attiva un customer con priorità maggiore
 - esegue una PRIOR che rende maggiore la priorità di un suo concorrente

Simulazione: Simul in Qnap2

- Note sul processo di simulazione
 - tutte le operazioni sono istantanee ad eccezione delle richieste di servizio (attive)
 - un customer rimane “attivo” dopo una transizione se arriva direttamente su un server disponibile e la sua priorità non decresce.
 - Una richiesta di servizio nulla è un trucco per forzare il passaggio di esecuzione di un customer verso altri al suo stesso livello di priorità.