

Pemodelan Sistem dan Simulasi

AL-ASWIR

Simulasi

- Simulasi: proses merancang model matematis atau logik dari sistem selanjutnya melakukan eksperimen dengan model tsb untuk menggambarkan, menjelaskan dan memprediksi kelakuan dari sistem
 - Kelebihan
 - Menunjukkan detail bisa dicakup
 - Dapat membandingkan rancangan sistem yang lain
 - Dapat mengontrol skala waktu
 - Sistem eksisting tidak diperlukan
 - Kelemahan
 - Sulit untuk menggeneralisir hasil
 - Sulit untuk mempertimbangkan semua nilai kasus/parameter
 - Sulit untuk menentukan sensitivitas
 - Waktu untuk mengembangkan dan mengeksekusi simulasi
 - Upaya untuk memvalidasi model dan menganalisa data output

Definisi-Definisi Sistem

- Definisi-Definisi Sistem
 - Parameter – kuantitas yang tetap (fixed) atau dp dikontrol
 - Variable – kuantitas yg ditentukan dari relasi-relasi fungsional
 - State Variables – set minimum dari variabel-variabel utk menggambarkan secara lengkap suatu sistem pd suatu titik dlm waktu
- Systems properties
 - Static – state variables independen terhadap waktu
 - Dynamic - state variables merupakan fungsi waktu
 - Continuous time – states merupakan fungsi kontinyu dari waktu
 - Discrete time – states hanya didefinisikan pd titik-titik waktu tertentu
 - Combined – sistem mengandung baik variabel-variabel kontinyu dan diskrit
 - Continuous State – state variables dp mempunyai harga dari suatu range kontinyu
 - Discrete state – state variables hanya mempunyai harga dari range diskrit
 - Deterministic - state variable dp diprediksi dg pasti
 - Stochastic – state variables mengandung sumber randomness

Definisi-Definisi Sistem

- Computer system dan communication networks adalah dynamic continuous time, discrete state, stochastic systems
- Cat: states hanya dp memp. harga dari suatu range diskrit dan konstan diantara perubahan
- Titik waktu dimana state berubah
 - suatu "event time"
- Apa yg menyebabkan state berubah disebut suatu "event"
 - Misalnya – kedatangan suatu packet ke suatu antrian router
- Sistem spt ini disebut Discrete Event Systems

Bagaimana Simulasi Berjalan (Programming Simulation)

- Bagaimana simulasi berjalan ...
 - Time-driven: simulasi berjalan pd interval waktu tertentu/fixed (mis. state ditentukan pada saat t , $t + \Delta t$, $t + 2 \Delta t$, ...)
→ Time-based simulation
 - Event-driven: simulasi berjalan dari event-ke-event (mis. state ditentukan pd titik waktu dari event berikutnya)
→ Event-based simulation

Time-Based Simulation

Event-Based Simulation

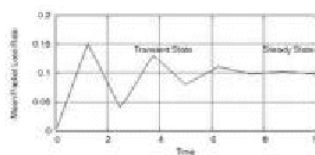
Mensimulasikan Discrete Event System

- **States**
Kumpulan variabel-variabel yg diperlukan utk karakterisasi sistem pada sembarang titik waktu
- **Entities**
Objek-objek yg diproses dalam simulasi – mis. packet atau panggilan telepon
- **Attributes**
Karakteristik dari entities (mis., panjang paket, tipe dan tujuan)
- **Resources**
Substansi/items dimana entities menduduki atau menggunakan (mis., buffer space pd router, tokens pd FDDI network, bandwidth pd suatu link)
- **Activities**
Durasi waktu dimana panjangnya diketahui saat dimulai. Misalnya, waktu transmisi dari suatu paket pd suatu link
- **Delay**
Durasi waktu dg panjang yg tdk terspesifikasi yg tdk diketahui sebelum selesai. misalnya – waktu perjalanan suatu paket dari node A ke node B dlm suatu jaringan

Mensimulasikan Kelakuan Sistem

- **Transient / Steady State**
 - Transient behavior : Suatu tipikal kelakuan sistem yg tergantung pd kondisi inisial (mis. booting up atau recovering dari suatu kegagalan komponen)
 - Steady state behavior : kelakuan operasi normal dari sistem independent terhadap kondisi inisial

Simulasi packet loss
pd suatu antrian
router



Event-Based Simulation (Event Scheduling Simulation)

- Utk discrete event systems, state dari sistem hanya berubah pd saat waktu event yg diskrit dlm titik waktu
- Cat: state sistem akan tetap tdk berubah diantara waktu event
- Sistem dp disimulaikan dg mulai dari tiap event dlm waktu ke event berikutnya dan menentukan efek apa dari event thd state sistem saat terjadi
- Utk mengimplementasikan suatu event scheduling simulasi memerlukan
 - Simulation Clock
 - Variabel memberikan nilai saat ini dari waktu yg disimulasikan
 - Event List
 - List dari waktu dan tipe dari tiap event masa depan/future event (diurut secara kronologis)

Event-Based Simulation (Event Scheduling Simulation)

- Pd event scheduled simulation – membangun model terdiri dari identifikasi
 - Entities:
 - Resources:
 - States
 - Events – dan efeknya pada state sistem
- Simulasi mencakup pergi dari event time ke event time dan menentukan efek pada state sistem – menghasilkan deretan snapshots dari kelakuan sistem
- Utk mengimplementasikan suatu event scheduling simulasi perlu membangun suatu event list, secara tdk langsung perlu membangun suatu deretan events
- Perhatikan contoh antrian single server – misalnya pada printer jaringan

Simulasi dg Tangan (Contoh)

- Perhatikan suatu sistem single server queueing
 - Entities : print jobs (customers)
 - Resources : printer (server of queue)
 - State sistem:
 - Jumlah dalam sistem (antrian + sedang diservice) $\Rightarrow n(t)$
 $n(t) = \{0, 1, 2, \dots\}$
 - Status printer $\Rightarrow sp(t)$
 $busy : sp = 1 \quad idle : sp = 0$
 - Events e_i :
 - Job arrival $\Rightarrow e_1 = 1$
 - Job departure $\Rightarrow e_2 = 2$

Efek Event pada State Sistem

```

/* arrival */
if ( e(i) = 1 ) then
  n = n+1;
  /* server idle */
  if ( sp = 0 ) then
    /* server busy */
    sp = 1;
  endif
endif

/* departure */
if ( e(i) = 2 ) then
  n = n-1;
  /* system empty */
  if ( n = 0 ) then
    /* server idle */
    sp = 0;
  endif
endif

```

Simulasi dg Tangan (Contoh)

- Menentukan input sistem – kita asumsikan waktu antar kedatangan dan waktu service adalah random dengan fungsi densitas di bawah
 - Job interarrival time
 $\{ 1, 2, 2.5, 4, 4.5, 5.5 \}$ dg equal prob sebesar $1/6$
 - Job service time (printing time)
 $\{ 1, 2, 3, 4 \}$ dg prob $1/4$
- Mendefinisikan variabel
 - t_i = time of arrival
 - IA_i = interarrival time antara job ke $(i-1)$ dan i
 - S_i = service time dari job ke i
- Dg membangkitkan deretan interarrival dan service times dari fungsi densitas di atas kita dp

Simulasi dg Tangan (Contoh)

Job i	IA_i	S_i
1	1	1
2	2	1
3	4	3
4	1	2
5	2.5	1
6	5.5	4

Simulasi dg Tangan (Contoh)

- Bangun event list dg mencari arrival times dan departure times utk tiap-tiap job
- Dapatkan
 - Waktu kedatangan (time of arrival)

$$t_i = t_{i-1} + IA_i$$
 - waktu job i meninggalkan sistem (selesai printing)

$$jd_i = bp_i + S_i$$
 - waktu job i mulai dp service (start printing)

$$bp_i = \max \{ t_i, jd_{i-1} \}$$

Membangun Event List

Job i	IA_i	t_i	bp_i	S_i	jd_i
1	1	1	1	1	2
2	2	3	3	1	4
3	4	7	7	3	10
4	1	8	8	2	10
5	2.5	10.5	10	1	11
6	5.5	16	16	4	20

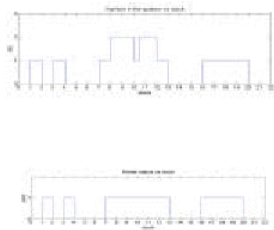
Event List

- Penyusunan kronologis dari event kedatangan dan kepergian bersamaan dg waktu/saat event – menghasilkan event list
- Simulasi berlangsung dg memproses event dari event list – menentukan efeknya pd state – kemudian bergerak ke waktu event berikutnya

Event List		State Variables	
Time	Event Type	n	sp
1	ar(1)	1	1
3	ar(2)	2	1
7	ar(3)	3	1
8	ar(4)	4	1
8	dp(4)	4	1
10	ar(5)	5	1
10	dp(3)	4	1
10.5	ar(5)	5	1
11	dp(5)	4	0
16	ar(6)	5	0
16	dp(6)	4	0

States

- Dari event list dp diplot state variabel vs waktu



Pengukuran Performansi

- Basis observasi
 - Delay dlm sistem → Response time
 - Delay dlm queue → Waiting time
 - dll,
 - Statistik dicari dg standar sample mean, variance, dll...
- Basis waktu
 - Jumlah pelanggan dlm sistem
 - Utilisasi server, dll.
 - Statistik dicari dg rata-rata waktu → integrasi

Pengukuran Performansi

- Contoh metrik performansi berbasis observasi
- Nilai rata-rata dari k pelanggan adalah:

$$W = \frac{1}{k} \sum_{i=1}^k d_i$$

dimana d_i = delay dari pelanggan ke- i

- Utk contoh single queue kita dapatkan:
 $W = 1/6[(2-1)+(4-3)+(10-7)+(12-8)+(13-10.5)+(20-16)] = 2.5833$

Pengukuran Performansi

- Contoh perata-rataan dlm waktu dari metrik performansi
- Jumlah rata-rata pelanggan dalam sistem(L) adalah:

$$L = \frac{1}{t_f} \int_0^{t_f} n(t) dt$$

- Utilisasi rata-rata dari server (ρ) adalah:

$$\rho = \frac{1}{t_f} \int_0^{t_f} sp(t) dt$$

Time Average Metric

- Cat: sulit utk melaksanakan integrasi numerik utk evaluasi metrik performansi – dihindari dg menggunakan kalkulasi geometrik sederhana
- Misalnya $n(t)$ adalah konstan diantara waktu event dan luas di bawah kurva dp ditentukan menggunakan penjumlahan persegi
- Misalkan clock, menunjukan waktu clock simulasi dari event ke- i

$$L = \frac{1}{t_f} \sum_i n(clock_{i-1})^+(clock_i - clock_{i-1})$$

dimana $()^+$ menyatakan mengambil sisi kanan diskontinuitas

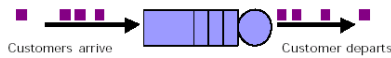
- Utk contoh single queue, didapatkan $t_f = 20$

$$L = 1/20[1(2-1)+1(4-3)+1(8-7)+2(10-8)+1(10.5-10)+2(12-10.5)+1(13-12)+1(20-16)] = 15.5/20 = .775$$

Event-Scheduled Computer Simulation

- Pertama, identifikasi event dan state yg diperlukan utk memodelkan sistem
 - Utk suatu sistem antrian → event kedatangan, event keberangkatan, state adalah jumlah dlm sistem dan status server
- Buat event list berisi waktu event yang di-schedule-kan terjadi di masa depan dlm urutan kronologis
 - $t \leq t_1 \leq t_2 \leq t_3 \dots \leq t_n$
- Naikkan clock utk event kedepan yg akan terjadi
- Pada tiap waktu event,
 - State diupdate utk event yg terjadi, sbg efek memberikan snapshot (gambaran) baru pd sistem yg dibangun
 - Event-event kedepan di-schedule → update event list (ini lebih baik drpd membangkitkan semua event disaat awal spt pd contoh dg tangan)
- Mudah diimplementasikan dg general purpose languages utk sistem yg kecil

Simulasi Antrian M/M/1



- Model
 - IID Exponential interarrival time dg mean $1/\lambda$
 - IID Exponential service time dg mean $1/\mu$
 - Infinite buffer
- Ingin mendapatkan mean customer delay dlm antrian
 - Solusi analitis $\rightarrow$ mean waiting time $= \rho / (\mu - \lambda)$
- Contoh code dlm bahasa C:

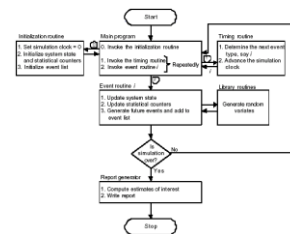
Komponen-Komponen Umum dlm Model Simulasi

1. State sistem
 - Variabel-variabel yg digunakan utk merepresentasikan variabel state
2. Simulation Clock
 - Variabel yg memberikan harga saat ini dari waktu yg disimulasikan
3. Event list
 - List dari waktu dan tipe dari tiap-tiap future event
4. Initialization routine
 - Subprogram utk inialisasi model simulasi pada awal dari tiap run dengan
 - 1) Men-set clock simulasi
 - 2) Men-set state sistem dan counter-counter statistik
 - 3) Schedule event pertama
5. Timing routine
 - Subprogram yg menentukan event berikutnya (next event) dari event list dan memajukan clock simulasi

Komponen-Komponen Umum dlm Model Simulasi

6. Event Routines
 - Subprograms (satu utk tiap tipe event) utk memproses event dg
 - Update system state
 - Update counter-counter statistik yg mungkin
 - Scheduled future events dari tipe yg sama (menentukan waktu event dan menambahkan ke event list)
7. Library routines
 - Set dari subprograms utk membangkitkan random variables dan mengumpulkan statistik
8. Report Generator
 - Subprogram yg menghitung statistik dan menghasilkan suatu report
9. Main program
 - Baca inputs schedules initialization, call event routines, report generator, end of program, dll...

Komponen-Komponen Model Simulasi



Common Features

- Mengimplementasikan suatu discrete event simulation dg general purpose language dipermudah dg memp. common features berikut
 - Random number generation
 - Harus uncorrelated, dlm prakteknya $u[0,1]$, cepat, reproducible.
 - Random variate generation
 - Menggunakan bilangan random sbg input
 - Inverse Transform method
 - Inisialisasi simulasi
 - Menaikkan clock
 - Kalkulasi waktu event, meneruskan kontrol diantara routines
 - Koleksi statistik dan analisis
 - Output formatting
 - Error debugging/traces

Software Simulasi

Secara kasar software tools utk membangun discrete event simulation dp dikategorikan kedlm empat kategori

1. General purpose languages
 - C, Pascal, FORTRAN, C++, ADA, Java, dll.
2. Event Scheduled Simulation Languages
 - SLAM, SIMAN, SIMPAS, SIM++, JAVASIM, dll,
3. Process Oriented Simulation Languages
 - CSIM, EZSIM, GPSS, SIMAN, SLAM, GASR, JAVASIM dll.
4. Application Oriented Simulators
 - Opnet, Connet III, Tangram II, ns-2, Qualnet, Jade, dll.

Membangkitkan Bilangan dan Variabel Random

- Random variables digunakan secara ekstensive pada discrete event simulation
- Perlu mampu utk membangkitkan sampel-sampel dari dari suatu random variable x dg suatu distribusi $F(x)$ yg dispesifikasikan
- Pendekatan umum utk melaksanakan ini:
 - Pertama-tama bangkitkan sampel-sampel dari suatu uniform zero - one random variable $U [0, 1]$
 - Transformasikan harga $U [0, 1]$ ke sampel-sampel dari random variable x .

Random Number Generator

- Random Number Generator (RNG)
 - Suatu algoritma yg akan menghasilkan deretan bilangan $U_1, U_2, U_3, \dots, U_k$ yg muncul dari sampel random variable $U[0,1]$
- Krn keterbatasan panjang word dari komputer
 - RNG akan selalu mengulangi deretan harga U_i setelah bbrp titik - disebut cycle atau perioda dari RNG
- Algoritma RNG adalah selalu deterministic
 - Orang biasa menggunakan istilah "pseudo-random number generator"

Properties dari RNG yg Baik

1. Bilangan-bilangan yg dibangkitkan lolos test statistik sbg distribusi $U[0,1]$
 - A chi-square test
2. Bilangan-bilangan yg dibangkitkan adalah independen
 - Bilangan random tdk berkorelasi
3. Reproducibility
 - Deretan yg sama akan selalu dibangkitkan utk suatu kondisi inisial yg spesifik
4. Komputasi efisien
5. Cycle time yg panjang antara repetisi dari bilangan-bilangan yg dibangkitkan
6. Memberikan hasil deretan multiple non-overlapping streams (sequences) dari bilangan-bilangan

Linear Congruential Generator (LCG)

- LCG didefinisikan oleh set berikut dari persamaan recursive,

$$Z_i = (a Z_{i-1} + c) \bmod m$$

$$U_i = Z_i / m$$
 - Z_0 : initial seed.
 - a : multiplier.
 - c : increment.
 - M : modulus.
- a, c, m dan seed Z_0 semuanya integer nonnegative dan memenuhi, $a < m, c < m, Z_0 < m$.

Contoh LCG

- LCG ($a=5, c=3, m=16, Z_0=7$)

i	Z_i	U_i
0	7	-
1	6	0.375
2	1	0.0625
3	8	0.5
4	11	0.6875
5	10	0.625
6	5	0.3125
7	12	0.75
8	15	0.9375

i	Z_i	U_i
9	14	0.875
10	9	0.5625
11	0	0
12	3	0.1875
13	2	0.125
14	13	0.8125
15	4	0.25
16	7	0.4375
17	6	0.375

- Mean = 0.4688, Variance = 0.0885

Karakteristi LCG

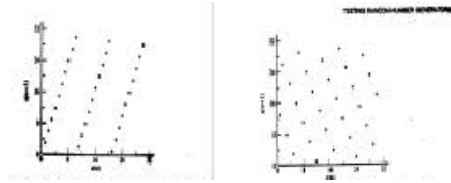
- Harga Z_i pd LCG didp dg operasi mod
 - $0 \leq Z_i \leq (m-1), \quad 0 \leq U_i \leq (m-1)/m$
- Harga U_i akan selalu dipilih dari set

$$U_i \in \left\{ \frac{0}{m}, \frac{2}{m}, \frac{3}{m}, \dots, \frac{m-1}{m} \right\}$$

- Harga m yg besar sebaiknya digunakan shg harga U_i solid dlm $(0,1)$
- Krn ada satu set terbatas dari harga U_i , LCG membangkitkan looping cycle.
 - Panjang dari cycle disebut perioda p dari generator
 - Secara umum, $0 \leq p \leq m$. Generator memp full period jika $p = m$

Generator LCG

- Cat: Banyak pilihan parameter LCG yg mungkin yg akan memberikan hasil dlm full period – tetapi tdk akan selalu memenuhi generator yg baik (mis., harga-harga tdk berkorelasi – perhatikan scatter plot dari dua LCG di bawah)



Standar Rekomendasi RNG

- Utk komputer 32-bit, rekomendasi standar minimum RNG adalah LCG ($a=75$, $c=0$, and $m=2^{31}-1$).
- Ini menghasilkan LCG :

$$Z_i = (16,807 Z_{i-1}) \bmod 2,147,483,647$$

$$U_i = \frac{Z_i}{2,147,483,647}$$

Initial Seeds

- Efek dari initial seed Z_0 adalah utk mengambil titik awal dalam cycle dari bilangan yang dihasilkan
 - Sbg contoh, LCG ($a=5$, $c=3$, $m=16$), jika seed $Z_0=8$ maka deretan yg sama didapat mulai dari entri ke empat dlm tabel sebelumnya
- Deretan-deretan dari non-overlapping random sequences dp dibangkitkan dg initial seeds yg berbeda
- Penggunaan deretan-deretan dari harga-harga yg independen
 - Membantu memutuskan korelasi dlm simulasi
 - Memberikan confidence intervals yg lebih akurat pada hasil

Random Variates

- Sampel-sampel dari suatu random variabel x dg distribusi $F(x)$ yg dispesifikasikan dp dibangkitkan dg berbagai metoda – salah satu pendekatan adalah
“the Inverse Transform technique”.
- Metoda Inverse Transform memetakan harga U_i melalui $F^{-1}(U_i)$ utk mendpkan harga X_i
- Algoritma Inverse Transform
 - Bangkitkan U_i dari suatu RNG $U[0,1]$
 - Set $x_i = F^{-1}(U_i)$.

Contoh Inverse Transform

- Asumsi x adalah terdistribusi eksponensial dg
 $F(x) = 1 - e^{-\lambda x}$ dan mean $= 1/\lambda$
- Tentukan fungsi inverse transform
 $U = F(x) = 1 - e^{-\lambda x}$
Selesaikan utk hasil-hasil x
 $X = (-1/\lambda) \ln(1-U) = -\text{mean} \times \ln(1-U)$
- Exponential random variables dibangkitkan dg
 - bangkitkan U_i dari suatu RNG $U[0,1]$
 - Set $x_i = -\text{mean} \times \ln(1-U_i)$

Discrete Random Variable

- Jika random variable x adalah discrete random variable dg harga $\{x_1, x_2, \dots, x_n\}$ dg $x_1 < x_2 < \dots < x_n$ dan densitas $P\{x = x_i\}$
- Maka, general inverse transform method algorithm:
 - Bangkitkan U_i dari suatu RNG $U[0,1]$
 - Cari j terkecil sedemikian shg $U_i \leq F(x_j)$.
 - Set $x_i = x_j$

Contoh

- Misalkan suatu random variable x dg distribusi
 $P(x=1)=0.2, P(x=2)=0.1, P(x=3)=0.3, P(x=4)=0.4$
- Algoritma utk membangkitkan random variable x adalah
 - 1) Bangkitkan U_i dari RNG $U[0,1]$
 - 2) If ($U_i \leq 0.2$) then
 $x_i = 1$
 elseif ($0.2 < U_i \leq 0.3$) then
 $x_i = 2$
 elseif ($0.3 < U_i \leq 0.6$) then
 $x_i = 3$
 else
 $x_i = 4$
 endif

General Purpose (GP) Language

- C, C++, Pascal, Fortran, dll.
- Keuntungan utama adalah modeler biasanya sudah menguasai satu language
- Dapat diakses secara universal pada setiap komputer
- Efisien dlm hal waktu eksekusi krn overhead implementasi yg lebih kecil
- Pemrograman yg lebih fleksibel
- Tetapi keuntungan menggunakan Simulation Language biasanya melebihi keuntungan drpd menggunakan general purpose language

Event Scheduled Simulation Language

- Commonality dari fitur-fitur yg diperlukan utk discrete event simulation (mis., pembangkitan random variables, event list, dll.) menyebabkan pengembangan event scheduled simulation languages (mis., Simpas, Sim++, SLAM, SIMAN, etc.)
- Event Scheduled Simulation Languages menyediakan framework utk event scheduled simulation
- Secara basic satu set library routines menyediakan common features yg diperlukan dlm discrete event simulation
 - Misalnya random variable generation, clock, event list, dll.
- Languages merupakan higher level general purpose language (mis., SIMPAS, Pascal), (SIM++, C++), (SLAM, Fortran), (SIMAN, Fortran)

Event Scheduled Simulation Language

- User menulis event routines dan main program dlm general purpose language utk memanggil library routines yg disediakan oleh simulation language
- Pros dan Cons
 - mengurangi lines dari code dan kesalahan dibangkitkan dg general purpose language
 - Kecepatan sebanding dg general purpose language
 - Memerlukan pengetahuan general purpose language dimana simulation language di-embedded-kan
 - Masih menulis banyak code – jika banyak events tercapak

Process-Oriented Simulation

- Suatu proses menggambarkan keseluruhan pengalaman dari "entity" saat "mengalir" melalui sistem
 - Bagaimana entities melalui sistem dan meninggalkan sistem
- Pendekatan lebih natural drpd event -scheduling
- Merepresentasikan suatu sistem dg suatu jaringan dari node-node diinterkoneksi dg cabang-cabang
- Node umumnya memodelkan proses yg terjadi pada discrete event systems
- Cabang-cabang memodelkan pergerakan entity

Process-Oriented Simulation

- Proses umum dlm discrete event systems
 1. Entity creation – bagaimana entities tiba ke system
 2. Entity termination - bagaimana entities dikeluarkan dari sistem
 3. Entity traversal – pergerakan entities melalui sistem
 4. Resources – substansi yg digunakan entities use atau dikonsumsi atau diduduki
 - Dua tipe resources: 1) service – server pd antrian, 2) regular – item yg dikonsumsi entities (token pd FDDI)
 5. Entity accumulation – dimana entities diantriakan
 6. Branch selection – proses routing entities dlm suatu sistem
 7. Entity multiplication/reduction – memodelkan cloning dari entities atau batching dari model-model

Process-Oriented Simulation

- Proses membangun model terdiri dari identifikasi entities dari sistem dan proses-proses yg dialaminya
- Model dibangun dg memilih node-node yg sesuai dg menghubungkan dg branches
- Selanjutnya parameterisasi nodes dan branches
- Event scheduling simulation dibangun dibawah model proses tetapi tersembunyi dari users
- Bbrp simulation languages mengimplementasikan pendekatan proses: SLAM, SIMAN, EZSIM, CSIM, GPSS, OPNET
- Cat: process oriented simulation languages umumnya memp lebih banyak node drpd tujuh proses yg umum (mis., node utk koleksi statistik, file/attribute manipulation, dll.)

Process-Oriented Simulation

Contoh Single Server Queue

- Entities: jobs
- Process: creation, accumulation, resource, termination
- Model



- Parameters
- Creation: waktu creation pertama = 0, waktu antar creation = exponential random variable dg mean = 1, attribute 1 = waktu creation
- Accumulation: FIFO queue, kapasitas tak terbatas (infinite)
- Resource: 1 server, service time = exponential dg mean = .5
- Termination: berhenti setelah 6000 jobs
- Branches: semua branches unconditional dg zero time delay

Process -Oriented Simulation Languages

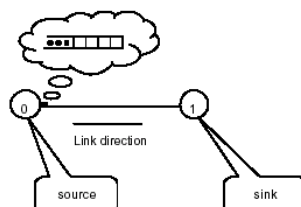
- Kebanyakan proses -oriented simulations memp. graphical user interface utk memudahkan model building dan parameter entering
 - Single server queue example in EZSIM
 - SLAM process model given on class web page process.dat
 - CSIM Process-oriented SL based on C language.
 - Example: csim.example on class web page
- Pros dan Cons
 - Menyediakan natural building blocks utk modeling.
 - Memerlukan sangat sedikit atau malah tdk memerlukan pemrograman
 - Dynamic resources allocation lebih baik
 - Less error – lebih sedikit, automatic error checking/identification.
 - Menyediakan struktur data yg memudahkan koleksi statistik
 - Kurang fleksibel dlm memodelkan sistem
 - Waktu eksekusi yg lebih lambat dg menggunakan prewritten precompiled blocks dari discrete event simulation code

Application Oriented Simulator

- Paket simulasi yg menyediakan software utk memodelkan suatu domain aplikasi
- Menyediakan prewritten event scheduled simulation utk memodelkan elemen-elemen common dlm suatu aplikasi tertentu (mis., routers, segmen ethernet, token rings, dll.)
- Umumnya mempunyai GUI
 - Representasi graphic dari sistem yg disimulasikan, animasi, data analysis tools
 - Sedikit atau tdk perlu upaya programming
 - Reusability dari software ? Cons
 - Konfigurasi terbatas, tidak fleksibel, akurasi model??
 - Mahal
- Opnet, Comnet III, Tangram II, Jade, Qualnet, ns-2

Contoh NS-2

- Dua node dg satu link dg antrian FIFO



- Akan dibahas pd kuliah berikutnya!

Tugas

(kumpul kuliah pertama setelah lebaran)

1. Consider a VoIP PBX telephone exchange at a corporation.

The PBX can connect up to 5 telephone calls simultaneously. The phone calls arrive to the PBX with an exponentially distributed amount of time between phone calls with a mean of 1 minute. The phone calls are of two types internal (80%) or external, 20%. Internal phone calls last an exponentially distributed amount of time with a mean of 3 minutes and long distance calls last an exponentially distributed amount of time with a mean of 10 minutes. The PBX can queue only 1 phone call if all 5 lines are busy, calls arriving to a full queue are dropped.

(a) Determine a 90% confidence interval with a relative precision of less than 7% on the utilization of the PBX.

(b) Find a 90% confidence interval on the percentage of local calls dropped. What is the relative precision?

You are free to use any language you want!.