PANDUAN PRAKTIKUM SIMULASI SISTEM INDUSTRI

Oleh : Arya Wirabhuana, S.T, M.Sc

Isi Paket :

Diktat Kuliah -
Studi Kasus -
Modul Praktikum -
Tugas Pendahuluan -
Software Simulasi -

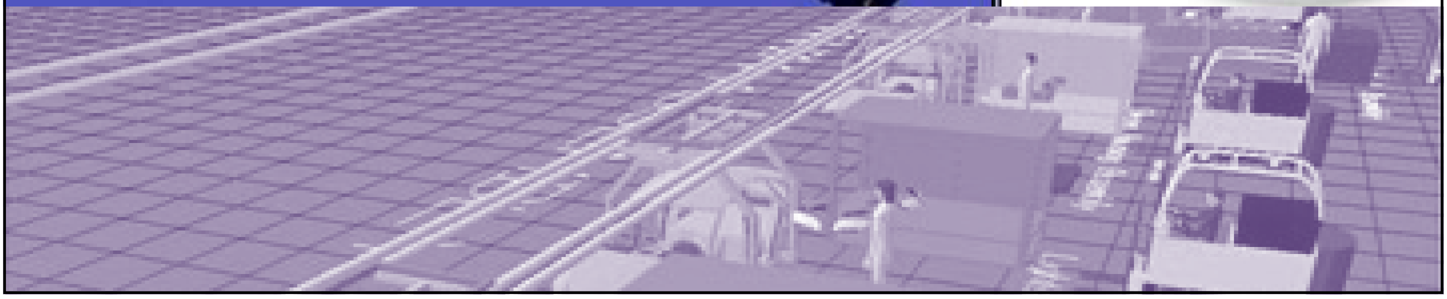
Pendamping Prakt :

M.Arief Rochman -
Agus Suwandi -
Medi Yanuarto –
Dwi Yuniarko -

Materi Praktikum :

1. *Pemodelan Sistem*
2. *Simulasi Monte Carlo dengan Ms.Excel*
3. *Web GPSS*
4. *ARENA*
5. *Automod*

Program Studi Teknik Industri
Fakultas Sains dan Teknologi
UIN Sunan Kalijaga
Tahun 2009



BAB. III.
PENGGUNAAN FUNGSI “BUILT-IN” PADA MS EXCEL

III.1. TUJUAN PRAKTIKUM

1. Memberikan pengenalan kepada mahasiswa mengenai fungsi - fungsi pada Ms. Excel yang sering digunakan pada simulasi.
2. Melatih mahasiswa untuk mengaplikasikan fungsi - fungsi pada Ms Excel yang sering digunakan pada penyelesaian masalah simulasi.

III.2. LANDASAN TEORI

III.2.1. Pendahuluan

Didalam perhitungan simulasi seringkali kita menggunakan fungsi – fungsi yang telah disediakan pada Excel pada Function Wizard. Fungsi adalah merupakan serangkaian formula rumit yang siap digunakan yang melakukan serangkaian operasi pada range nilai tertentu.

Setiap fungsi terdiri dari tiga elemen berikut ini:

- Tanda sama dengan (=) menandakan bahwa yang mengikuti adalah sebuah fungsi / formula.
- Nama fungsi, seperti IF, menandakan operasi mana yang akan dilakukan.
- Argumen, seperti (A1:H1), menandakan alamat sel dari nilai yang akan digunakan fungsi. Argumen kadang berupa range sel, tetapi bisa juga lebih rumit.

Adapun fungsi – fungsi Excel yang sering digunakan antara lain adalah:

III.2.2. Logika.

1. **AND(*logical1*, logical2,…..)**

   Operasi ini menghasilkan TRUE (benar) jika semua argumen logika adalah TRUE, sebaliknya FALSE (salah) jika semua argumen logika adalah FALSE.

Contoh:

=AND (B36;C12>20) adalah TRUE hanya jika B36 tidak nol dan C12 lebih dari 20.

2. **IF*(logical_test, value_if_true,value_if_false)***

   Menentukan suatu tes logika untuk dikerjakan dengan pengandaian.

   Contoh:

   =IF(A3>=100,A3*2,A2*2) adalah jika A3 lebih besar atau sama dengan 100, formula A3*2 digunakan, tetapi jika A3 lebih kecil dari 100, formula A2*2 yang digunakan.

3. **NOT*(logical)***

   Operasi logika ini membalik argumen logika. TRUE ke FALSE, FALSE ke TRUE.

   Contoh :

   =IF(NOT(OR(B36=12,B36=20));"Bukan 12 atau 20";"12 atau 20")

   Pernyataan terakhir dihasilkan bila B36 berisi 12 atau 20 dan menghasilkan pesan berbunyi Bukan 12 atau 20 bila B36 berisi angka yang lain.

4. **OR(*logical1*, logical2,…..)**

   Operasi logika ini menghasilkan TRUE jika satu atau lebih argumen logika adalah TRUE, dan FALSE hanya jika semua argumen logika adalah FALSE.

   Contoh:

   IF(OR(B36=12,B36=20);"12 atau 20";"Bukan 12 atau 20)

   Pernyataan ini memeriksa apakah sel B36 berisi bilangan 12 atau 20, bila ya akan ditampilkan tulisan 12 atau 20. Begitu pula sebaliknya.

III.2.3. Statistik.

1. **AVARAGE (*number1*, number2,….)**

   Menghasilkan nilai rata – rata dari sejumlah data atau range data.

   Contoh:

   =AVARAGE(A1:A4) menghasilkan nilai rata – rata 2.5 bila sel A1 sampai A4 berisi 1,2,3 dan 4.

2. **STDEV (*number1*, number2,….)**

   Menghitung standar deviasi dari suatu populasi berdasar sampel populasinya.

   Contoh:

   =STDEV(A1:A11) menghasilkan 12.12 bila range dari A1:A11 berturut – turut berisikan 98, 67, 89,76, 76, 54, 87, 78, 85, 83, dan 90.

3. **VAR (*number1*, number2,….)**

   Menghitung variansi suatu populasi berdasar sampel populasinya.

   Contoh:

   =VAR(A1:A11) menghasilkan 12.12 bila range dari A1:A11 berturut – turut berisikan 98, 67, 89,76, 76, 54, 87, 78, 85, 83, dan 90.

4. **FREQUENCY(*data_array*;bins_array)**

   Menghasilkan distribusi frekuensi sebagai array vertikal.

   Contoh:

   =FREQUENCY(A1:A9,C4:C6) menghasilkan {0;2;5;2} bila nilai pada A1:A9 adalah 79, 85, 78, 85, 83, 81, 95, 88, dan 97. Nilai pada C4:C6 yaitu 70, 79, 89. Ketika FREQUENCY dimasukkan sebagai array, nomer dari nilai korespodensi pada tingkat range 0-70, 71-79, 80-89, dan 90-100, adalah terhitung. Contoh ini mengasumsikan semua nilai tes adalah integer. Formula tersebut dimasukkan sebagai sebuah rumus array setelah anda memilih empat sel vertikal adjancent pada data anda.

III.2.4. Matematika dan Trigonometri

1. **SUM (*number1*, number2,….)**

   Menghitung total pada range tertentu.

   Contoh:

   =SUM(A1:A5) menghasilkan 18 bila range dari A1:A5 berturut – turut 3,4,1,8,2.

2. **SQRT(number)**

   Menghasilkan akar kuadrat positif dari angka yang dimasukkan.

   Contoh:

   =SQRT(25) menghasilkan 5.

3. **SUMIF(*range,criteria,*sum_range)**

   Menjumlahkan sel didalam range bila sesuai dengan kriteria yang ditetapkan.

   Contoh:

   =SUMIF(A1:D1;”>50”) akan menghasilkan 150 bila range A1:D1 berisi nilai 75,25,75,33.

4. **ROUND(*number;*num_digits)**

   Membulatkan a bilangan menjadi ke jumlah digit tertentu.

   Contoh:

   =ROUND (2.15, 1) menghasilkan 2.2

   =ROUND(2.149, 1) menghasilkan 2.1

   =ROUND(-1.475, 2) menghasilkan -1.48

   =ROUND(21.5, -1) menghasilkan 20

5. **INT(number)**

   Membulatkan ke bawah, ke bilangan bulat terdekat.

   Contoh:

   =INT(8.9) menghasilkan 8

   =INT(-8.9) menghasilkan -9

III.3. TUGAS PENDAHULUAN

Buatlah sebuah ikhtisar/ringkasan yang berisi tentang penjelasan cara penggunaan fungsi “Built-in” yang ada pada Ms. Excel yang terdiri dari:

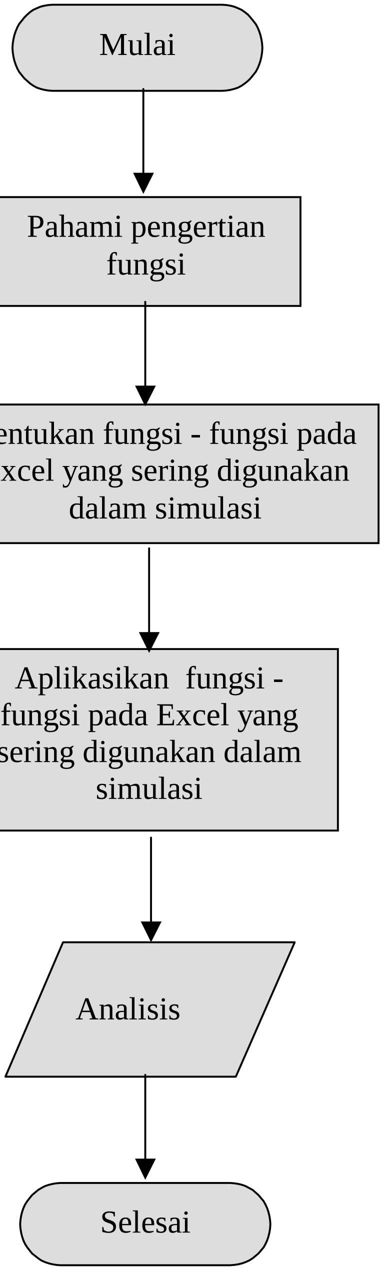
- Fungsi Statistika : Minimal 8 Fungsi
- Fungsi Matematika : Minimal 8 Fungsi
- Fungsi Logika : Minimal 4 Fungsi

Ringkasan ditulis tangan pada kertas HVS folio Bergaris

III.4. PROSEDUR PRAKTIKUM

1. Pelaksanaan Pre-Test
2. Penjelasan materi dengan metode interaktif
3. Contoh Penggunaan fungsi “Built-in”

III.5. FLOW CHART PRAKTIKUM


Mulai
Pahami pengertian fungsi
Tentukan fungsi - fungsi pada Excel yang sering digunakan dalam simulasi
Aplikasikan fungsi - fungsi pada Excel yang sering digunakan dalam simulasi
Analisis
Selesai

BAB. IV
APLIKASI MS EXCEL PADA SIMULASI SISTEM
KASUS I – ANTRIAN TUNGGAL

IV.1. TUJUAN PRAKTIKUM

1. Memperkenalkan penggunaan software Microsoft Excel pada Penyelesaian masalah Antrian Tunggal
2. Dapat menggunakan fungsi-fungsi pada Mis Excel guna penyelesaian model simulasi
3. Dapat melakukan interpretasi dari hasil Simulasi yang dilakukan
4. Memiliki kemauan untuk mengembangkan hasil yang telah ada.
5. Menumbuhkan kreatifitas dalam penggunaan Ms. Excel pada masalah-masalah Simulasi lainnya.

IV.2. LANDASAN TEORI

IV.2.1 Teori Antrian

IV.2.1.1. Pendahuluan

Dalam kehidupan sehari – hari kata antrian (*queuing*) atau *waiting line* sangat sering kita jumpai. Istilah antrian digunakan untuk menggambarkan sejumlah kelas gejala (fenomena) yang memuat kedatangan, penantian, layanan, dan keberangkatan.

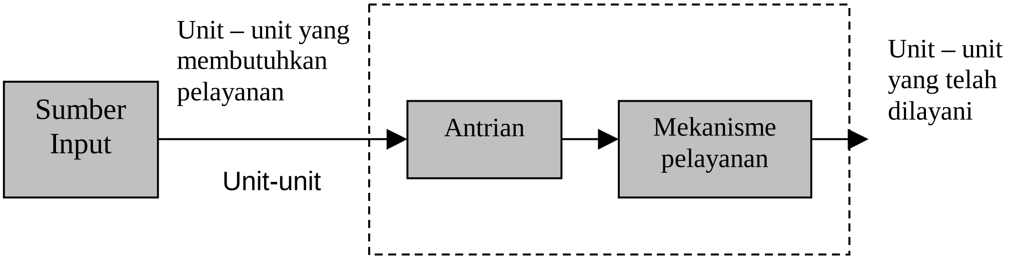
Teori tentang antrian diketemukan dan dikembangkan oleh A.K. Erlang, seorang insiyur dari Denmark yang bekerja pada perusahaan telepon di Kopenhagen pada tahun 1910. Dia melakukan eksperimen tentang fluktuasi permintaan fasilitas telepon yang berhubungan dengan *automatic dialing equipment*, yaitu peralatan penyambungan telepon secara otomatis. Dalam waktu–waktu yang sibuk operator sangat kewalahan untuk melayani para penelpon secepatnya, sehingga para penelpon harus antri menunggu giliran yang cukup lama. Persoalan aslinya Erlang hanya memperlakukan perhitungan keterlambatan (*delay*) dari seorang operator, kemudian pada tahun 1917 studi atau penelitian dilanjutkan untuk menghitung kesibukan beberapa operator.

Dalam periode ini dia menerbitkan bukunya yang terkenal berjudul *Solution of some problems in the theory of probabilities of significance in Automatic Telephone Exchange*. Baru setelah perang dunia kedua, hasil penelitian Erlang diperluas lagi penggunaanya antara lain dalam teori antrian (*queues*).

Teori antrian adalah teori yang menyangkut studi matematis dari antrian – antrian atau baris – baris penungguan. Formasi baris – baris penungguan ini tentu saja merupakan suatu fenomena biasa yang terjadi apabila kebutuhan akan suatu pelayanan melebihi kapasitas yang tersedia untuk menyelenggarakan pelayanan itu.

<u>IV.2.1.2. Struktur dasar model antrian</u>

Proses yang terjadi pada model antrian dapat digambarkan sebagai berikut:



Gambar IV.1. Suatu Model Diagram Sistem Antrian

Unit – unit yang memerlukan pelayanan yang diturunkan dari suatu sumber input memasuki sistem antrian dan ikut dalam antrian. Dalam waktu-waktu tertentu , anggota antrian ini dipilih untuk dilayani. Pemilihan ini didasarkan pada suatu aturan tertentu yang disebut **disiplin antrian** atau *service dicipline*. Pelayanan yang diperlukan dilaksanakan dengan suatu **mekanisme pelayanan** tertentu atau *service machanism*. Setelah itu, unit-unit tersebut meninggalkan sistem antrian.

<u>IV.2.1.3. Sumber Input</u>

Suatu karakteristik yang perlu diketahui dari sumber input ini adalah ukuran atau jumlahnya, yaitu total unit yang memerlukan pelayanan dari waktu ke waktu atau disebut jumlah total unit-unit potensial. Ini bisa dianggap terbatas ataupun tidak terbatas. Karena perhitungannya akan lebih mudah untuk jumlah unit yang tidak terbatas, maka asumsi ini sering digunakan, terlebih lagi jika unit ini cukup besar. Untu jumlah unit-unit yang terbatas, perhitungan akan lebih sulit lagi karena jumlah unit dalam sistem antrian akan mempengaruhi jumlah unit-unit potensial di luar sistem setiap waktu. Bagaimanapun, asumsi jumlah yang terbatas ini tetap harus dibuat jika sumber input yang menurunkan atau menghasilkan unit-unit yang memerlukan pelayanan ini jelas-jelas dipengaruhi oleh jumlah unit dalam sistem antrian.

Pola statistik dari penurunan unit-unit yang memerlukan pelayanan ini harus juga ditentukan. Dalam hal ini, asumsi yang biasa digunakan adalah unit-unit ini diturunkan dengan mengikuti proses Poisson, artinya sampai suatu waktu tertentu jumlah unit yang diturunkan ini mempunyai distribusi Poisson. Ini adalah suatu kasus dimana kedatangan pada sistem antrian terjadi secara random, tetapi dengan tingkat rata-rata tertentu. Asumsi berikutnya adalah bahwa distribusi kemungkinan dari waktu antarkedatangan (*inter-arrival time*) adalah distribusi Eksponensial.

Asumsi lain yang juga harus dispesifikasikan mengenai kelakuan unit-unit yang memerlukan pelayanan ini adalah apa yang disebut *balking*, yaitu bahwa unit-unit yang memerlukan pelayanan itu akan menolak memasuki sistem antrian jika antrian itu terlalu panjang.

IV.2.1.4. Antrian

Karakteristik suatu antrian ditentukan oleh jumlah unit maksimum yang boleh ada di dalam sistemnya. Antrian ini dikatakan terbatas atau tidak terbatas, bergantung pada apakah jumlah unitnya terbatas atau tidak terbatas.

IV.2.1.5. Disiplin Pelayanan

Disiplin pelayanan berkaitan dengan cara memilih anggota antrian yang akan dilayani. Sebagai contoh disiplin pelayanan ini berupa yang datang lebih dahulu dilayani lebih dahulu (*Fist Come Fist Serve*), random, atau dapat pula berdasarkan prosedur prioritas tertentu. Jika tidak ada keterangan apa-apa tentang disiplin pelayanan ini, maka asumsi yang biasa digunakan adalah *Fist Come Fist Serve*

IV.2.1.6. Mekanisme Pelayanan

Mekanisme pelayanan terdiri dari satu atau lebih fasilitas pelayanan yang masing – masing terdiri atas satu atau lebuih saluran pelayanan paralel. Jika ada lebih dari satu fasilitas pelayanan, maka unit-unit yang memerlukan pelayanan akan dilayani oleh serangkaian fasilitas pelayanan ini. Suatu model antrian harus menetapkan urut-urutan fasilitas semacam itu sekaligus dengan jumlah pelayanan pada masing-masing saluran paralelnya. Kebanyakan model –model dasar mengasumsikan satu fasilitas pelayanan dengan satu atau beberapa pelayan (terbatas).

Waktu yang digunakan sejak pelayanan dimulai sampai satu unit selesai dilayani, disebut sebagai waktu pelayanan (*holding time*). Biasanya diasumsikan bahwa distribusi kemungkinan dari waktu pelayanan ini adalah distribusi Erlang atau distribusi Eksponensial atau waktu pelayanan tetap (*constant service time*).

IV.2.1.7. Model Single Server (S=1)

Model – model antrian dapat mempunyai pelayanan tunggal, dapat pula mempunyai jumlah pelayan yang banyak. Berikut ini adalah model – model antrian yang digunakan apabila pelayannya hanya ada satu (S=1) dengan disiplin pelayanan tertentu.

Dengan input Poisson dan waktu pelayanan Eksponensial maka model ini dengan $\lambda_n = \lambda$ untuk n = 0,1,2,3….dan $\mu_n = \mu$ untuk n = 0,1,2,3…. dan dari *steady state* ($\lambda < \mu$) kita peroleh :

$$Po = \frac{1}{1 + \sum_{n=1}^{\infty} \frac{\pi_{i=0}^{n-1} \lambda i}{\pi_{i=1}^{n} \mu i}} = \frac{1}{\sum_{n=0}^{\infty} \left(\frac{\lambda}{\mu} \right)^{n}}$$

$$= \left(\frac{1}{1 - \lambda/\mu} \right)^{-1} = 1 - \lambda/\mu$$

Untuk n > 0

$$Pn = Po \frac{\pi_{i=0}^{n-1}}{\pi_{i=1}^{n} \mu i} = Po\left(\lambda/\mu \right)^{n}$$

Karena $\rho = \lambda/\mu$, maka :

$$\boxed{Pn = (1 - \rho)\rho^{n}}$$ Untuk n = 0,1,2,3….

Dengan demikian, maka :

$$L = \sum_{n=0}^{\infty} n(1 - \rho)\rho^{n} = (1 - \rho)\rho \sum_{n=0}^{\infty} \frac{d}{d\rho}\left(\rho^{n}\right)$$

$$= (1 - \rho)\rho \frac{d}{d\rho}\left(\sum_{n=0}^{\infty} \rho^{n} \right) = (1 - \rho)\rho \frac{d}{d\rho}\left(\frac{1}{1 - \rho} \right)$$

$$\boxed{L = \frac{\rho}{1 - \rho} = \frac{\lambda}{\mu - \lambda}}$$

Dengan cara yang sama :

$$Lq = \sum_{n-1}^{\infty} (n - 1)Pn$$

$$= L - 1(1 - P_{o})$$

$$\boxed{Lq = \frac{\lambda^{2}}{\mu(\mu - \lambda)}}$$

Masih dengan asumsi $\lambda \succ \mu$, kita akan mencari untuk waktu menunggu.

$$\boxed{W = \frac{1}{\mu(1-\rho)} = \frac{1}{\mu - \lambda}}$$

Pada beberapa kasus, waktu menunggu yang lebih relevan adalah waktu menunggu yang hanya sampai pelayanan dimulai. Jika dalam sistem telah ada n = 0 unit – unit, maka ia harus menunggu selama n waktu pelayanan eksponensial, hingga ia mulai dilayani, sehingga :

$$\boxed{Wq = \frac{\lambda}{\mu(\mu - \lambda)}}$$

Keterangan:

E_n : keadaan dimana ada n *calling unit* dalam sistem antrian.

$P_n(t)$: kemungkinan bahwa tepat ada n *calling unit* pada sistem antrian pada saat t.

S : jumlah *server* pada sistem antrian.

λ_n : tingkat kedatangan rata – rata dari *calling unit* baru jika ada n unit dalam sistem.

μ_n : tingkat pelayanan rata – rata jika ada n unit dalam sistem.

P_n : kemungkinan bahwa tepat ada n *calling unit* dalam sistem antrian.

L : ekspektasi sistem antrian.

L_q : ekspektasi panjang antrian.

W : ekspektasi waktu menunggu dalam sistem.

Wq : ekspektasi waktu menunggu dalam antrian.

II.2.1.8. Model Multiple Server (S > 1)

Dengan input Poisson dan waktu pelayanan eksponensial, model ini mengasumsikan bahwa kedatangan terjadi menurut input Poisson dengan parameter λ, dan bahwa waktu pelayanan untuk masing – masing unit mempunyai distribusi eksponensial dengan rata – rata $1/\mu$. Jadi distribusi waktu pelayanan sama, tanpa memperhatikan pelayanan mana dari sejumlah S pelayan yang melakukan pelayanan untuk unit.

Tingkat pelayanan rata – rata untuk seluruh sistem antrian adalah tingkat rata – rata dimana unit yang sudah dilayani meninggalkan sistem, dan bergantung pada *state* sistem En. Tingkat pelayanan rata – rata per pelayanan yang sibuk adalah μ, karena itu tingkat pelayanan keseluruhan adalah $\mu_n = n\mu$ jika n $\leq s$. Jika $n \geq s$, berarti semua pelayan sibuk sehingga $\mu_n = S\mu.$

Jika $\lambda \prec S\mu$ maka hasil *steady state*-nya adalah:

$$Po = \frac{1}{\sum_{n=0}^{s-1}\frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{S!}\sum_{n=s}^{\infty}\left(\frac{\lambda}{S\mu}\right)^{n-s}}$$

$$Po = \frac{1}{\sum_{n=0}^{s-1}\frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{S!}\frac{1}{1-\lambda/S\mu}}$$

dan

$$\frac{(\lambda/\mu)^n}{n!}Po \qquad \text{jika } 0 \leq \text{n} \leq \text{S}$$

Po =

$$\frac{(\lambda/\mu)^n}{S!S^{n-s}}Po \qquad \text{jika n} \geq \text{S}$$

Dengan $\rho = \lambda/S\mu$, maka :

$$L_q \quad \frac{Po\;(\lambda\;/\mu\;)^s\;\rho}{S\;!\;(1\;-\;\rho\;)^2}$$

$$W_q = \frac{L_q}{\lambda}$$

$$W = W_q + \frac{1}{\mu}$$

$$L = \lambda\left(W_q + \frac{1}{\mu}\right) = L_q + \frac{\lambda}{\mu}$$

IV.2.2. Pendekatan Simulasi Sistem pada Masalah Antrian

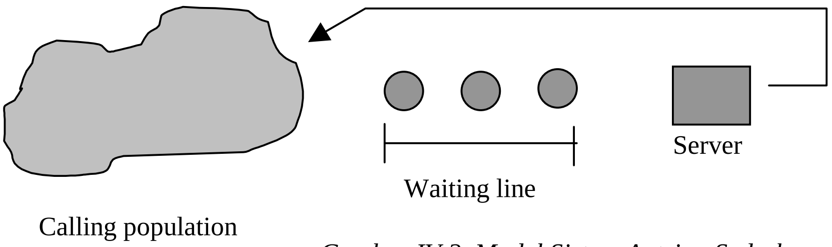
IV.2.2.1. Pendekatan Simulasi

Simulasi, adalah sebuah pendekatan yang saat ini sering digunakan untuk menganalisis sebuah sistem antrian. Sebuah proses yang sangat spesifik pada antrian adalah orang datang, memasuki sebuah *waiting line*, menunggu , dilayani, dan pada akhirnya meninggalkan sistem. Pendekatan simulasi agaknya memiliki bebrapa kelebihan dalam penerapannya sebagai alat / tools untuk menganalisis sebuah sistem antrian. Selain metode ini memungkinkan pengamat menganilisis sebuah sistem dengan lebih analitis, metoda pengamtan dengan simulasi juga dapat digunakan sebagai sebuah representasi dari sistem nyata yang cukup baik. Hingga jika kita akan melakukan eksperimen-0eksperimen guna mengubah sistem nyata untuk peningkatan performasndinya, dapat dilakukan dengan melakukan perubahan-perubahan pada model simulasi yang tidak memiliki resiko tinggi dalam implementasi perubahan-perubahan yang dilakukan.

Dalam kenyataannya, banyak hal yang dapat didekati dengan konsep sistem antrian. Beberapa fasilitas-fasilitas pelayanan, sistem-sistem produksi, perbaikan dan fasilitas-fasilitas perawatan, komunikasi dan sistem komputer, dan transport serta sistem pemindahan material dapat digambarkan sebagai sistem antrian.

Sebuah sistem antrian dapat dideskripsikan sebagai apa yang disebut dengan populasi, sifat dari kedatangan-kedatangan dan pelayanan-pelayanan, kapasitas sistem, dan disiplin antrian.

Sebuah sitem antrian yang sederhana digambarkan sebagai berikut :



Gambar IV.2. Model Sistem Antrian Sederhana

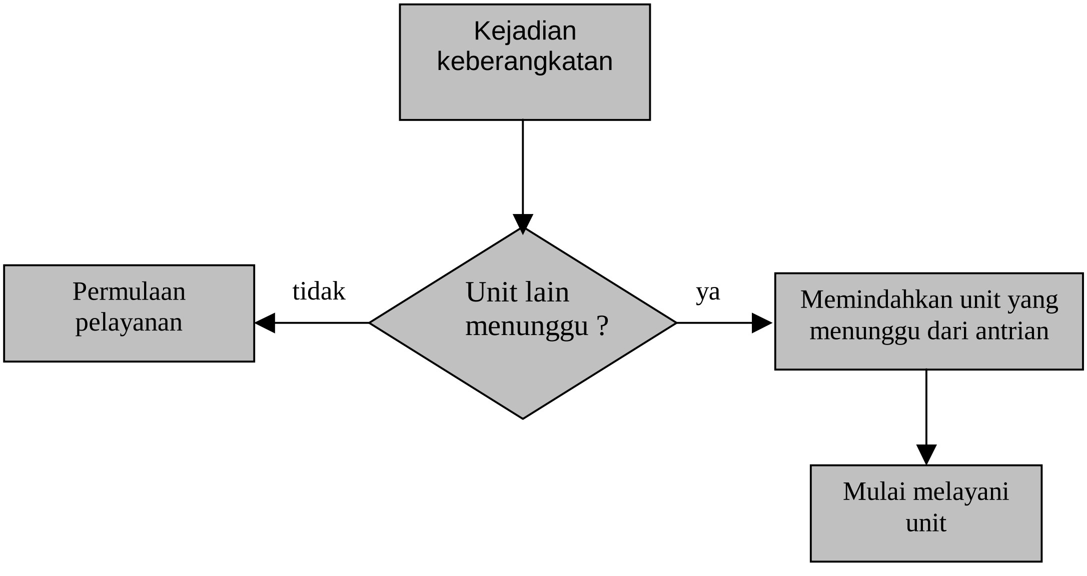
Di dalam sistem ini apa yang disebut populasi dapat diasumsikan menjadi finitif atau infinitif . Sebagai contoh, mempertimbangkan sebuah bank lima mesin yang memperbaiki ban-ban. Setelah selang waktu tertentu, sebuah mesin secara otomatis membuka dan harus diurus oleh seorang pekerja yang memindahkan ban dan menaruh ban tersebut ke dalam mesin. Mesin-mesin adalah konsumen. Pekerja adalah server. Dalam sebuah sistem dengan populasi yang besar dengan konsumen potensial, biasanya diasumsikan infinitif. Contoh, konsumen potensial dalam restoran, bank dan fasilitas-fasilitas lain yang serupa. Bisa juga, jika satu unit meninggalkan populasi dan bergabung ke jalur antrian atau memasuki pelayanan, maka tidak ada perubahan dalam tingkat kedatangan dari unit-unit lain yang mungkin membutuhkan pelayanan. Juga, didalam sistem ini, kedatangan ke pelayanan terjadi sekali pada suatu waktu di dalam mode random dan sekali waktu bergabung dalam jalur antrian, mereka tetap dilayani pada akhirnya. Sebagai tambahan, waktu pelayanan adalah panjangnya bilangan random yang mengikuiti sebuah distribusi probabilitas yang tidak mengalami perubahan. Juga, kapasitas sistem tidak terbatas. Pada akhirnya, yang terpanggil dilayani sesuai kedatangannya (sering disebut FIFO) oleh single server atau bercabang/multi server.

Kedatangan dan pelayanan dideskripsikan oleh distribusi waktu antar kedatangan dan waktu pelayanan. Tingkat kedatangan efektif secara keseluruhan harus lebih rendah dari tingkat pelayanan maksimum, atau jalur menunggu akan meningkat tanpa batas. Ketika antrian meningkat tanpa batas, maka disebut ‘ledakan’ atau tidak stabil.

Perbedaan utama antara finitif dan model populasi infinitif adalah bagaimana tingkat kedatangan didefinisikan. Pada model populasi infinitif, tingkat kedatangan tidak dipengaruhi oleh jumlah konsumen yang meninggalkan populasi dan bergabung ke sistem antrian. Ketika proses kedatangan homogen dalam waktu yang lama (disini tidak ada “rush hours”), tingkat kedatangan biasanya diasumsikan konstan. Dilain pihak, untuk model populasi finitif, tingkat kedatangan ke sistem antrian tergantung pada jumlah konsumen yang akn dilayani dan yang menunggu.

Prioritas untuk memperkenalkan beberapa simulasi sistem antrian, diperlukan pemahaman tentang konsep-konsep bagian sistem, kejadian-kejadian, dan waktu jam. Bagian dari sistem adalah jumlah unit-unit di dalam sistem dan status dari pelayan, sibuk atau menganggur. Di dalam sebuah sistem antrian tunggal hanya ad dua kemungkinan kejadian yang dapat mempengaruhi bagian dari sistem. Yaitu masukan dari suatu unit ke dalam sistem (kejadian kedatangan) atau penyelesaian pelayanan pada suatu unit . Sistem antrian terdiri atas server, unit yang akan dilayani , dan unit-unit dalam antrian.

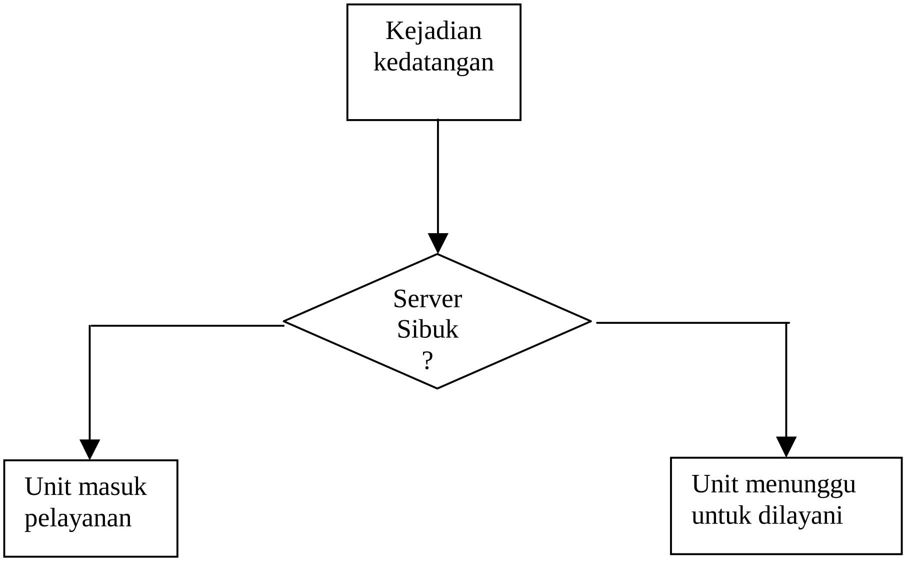
Jika pelayanan baru saja dilengkapi, hasil - hasil simulasi dalam suatu cara diperlihatkan pada flow diagram pada gambar 2 . sebagai catatan, server hanya mempunyai dua kondisi yang mungkin; sibuk atau menganggur.



Gambar IV.3. Model Diagram prosedur kerja sebuah server

Kejadian kedua terjadi ketika sebuah unit memasuki sistem. Diagram alir terlihat pada gambar .

Unit mungkin menemukan server baik sibuk atau menganggur; oleh karena itu, baik unit memasuki server, atau memasuki antrian menunggu pelayanan. Unit mengikuti cara kerja seperti yang ditunjukkan pada gambar II..4. Jika server sibuk, unit memasuki antrian. Jika server menganggur dan antrian dalam keadaan kosong, unit masek ke pelayanan. Hal tersebut tidak mungkin untuk server menganggur dan antrian tidak kosong.



		Status antrian	
		Tidak kosong	Kosong
Status pelayanan	sibuk	Masuk antrian	Masuk antrian
	Menganggur	Tidak mungkin	Masuk pelayanan

Gambar IV.4. Prosedur Kejadian Unit Datang

Setelah menyelesaikan pelayanan, server kemungkinan menjadi menganggur, atau kembali sibuk dengan unit berikutnya. Hubungan keduanya ditunjukkan dalam gambar berikut :

		Status antrian	
		Tidak kosong	Kosong
Status pelayanan	Sibuk	▨	Masuk antrian
	Menganggur	Tidak mungkin	▨

Gambar IV.5. Kemungkinan Hubungan Status Pelayanan dan Status Antrian

Sekarang, bagaimana kejadian-kejadian dapat dideskripsikan pada kejadian dalam waktu pensimulasian ? simulasi-simulasi sistem antrian pada umumnya membutuhkan perawatan daftar kejadian untuk menentukan apa yang akan terjadi kemudian. Daftar kejadian mengindikasikan waktu pada saat tipe-tipe yang berbeda dari kejadian terjadi untuk masing-masing unit dalam sistem antrian. Dalam simulasi, kejadian-kejadian biasanya terjadi secara random. Kerandoman mencontohkan dunia nyata untuk melukiskan ketidaktentuan. Sebagai contoh, tidak diketahui secara pasti kapan pelanggan berikutnya akan datang pada sebuah toko pangan dan meninggalkan kounter, atau berapa panjang teller bank akan dapat menyelesaikan sebuah transaksi.

IV.2.2.2 Karakteristik Sistem Antrian

Elemen-elemen kunci sistem antrian adalah konsumen dan server-server. Bentuk konsumen dan diumpamakan orang, mesin, truk-truk, mekanik, pasien, pesawat, kasus-kasus, permintaan, dan lainnya.

Bentuk server bisa menunjukkan resepsionis, mekanik, personel kesehatan mesin pencuci dan lainnya yang menyediakan pelayanan. Meskipun terminologi dikerjakan dimana konsumen mendatangi fasilitas pelayanan, kadang-kadang srver bergerak ke konsumen. Sebagai contoh, Mekanik mendatangi mesin yang rusak.

Pada tabel berikut memperlihatkan beberpa sistem yang berbeda.

Sistem	Konsumen	Server (s)
Meja resepsionis	Orang-orng	Resepsionis
Fasilitas perbaikan	Mesin-mesin	Tukang reparasi
Garasi	Truk	Mesin-mesin
Rumah sakit	Pasien	Perawat
Jaringan jalan	Mobil	Traffic light
Computer	Tugas-tugas	CPU, disket
Bank	Nasabah	Teller

Tabel IV.1. Contoh beberapa sistem antrian

IV.2.2.3. Apa yang disebut Populasi

Seperti telah dijelaskan sebelumnya, apa yang disebut dengan populasi bisa diasumsikan sebagai bentuk **infinitif** dan **finitif**.. Model populasi infinif tingkat kedatangannya tidak terpengaruh oleh jumlah konsumen yang meninggalkan populasi dan bergabung dalam sistem antrian. Sedangkan model finitif, tingkat kedatangan pada sistem antrian tergantung pada jumlah konsumen yang akan dilayani dan yang menunggu.

IV.2.2.4. Kapasitas sistem

Di dalam beberapa sistem-sistem antrian disana ada suatu batas untuk jumlah konsumen yang mungkin ada dalam jalur menunggu atau sistem.. Sebagai contoh,pembersih mobil otomatis mempunyai ruangan hanya untuk 10 mobil yang menunggu di jalan untuk masuk dalam mekanisme. Itu sangat mungkin berbahaya atau ilegal untuk mobil-mobil yang menunggu di jalan. Beberapa sistem, sperti penjualan tiket konser untuk pelajar-pelajar, mungkin dipertimbangkan mempunyai kapasitas yang tidak terbatas.. Disana tidak ada batas pada jumlah pelajar yang menunggu untuk membeli tiket.

IV.2.2.5. Proses kedatangan.

Kedatangan mungkin terjadi pada waktu yang ditetapkan atau pada waktu random. Pada saat waktu random, waktu antar kedatangan biasanya mempunyai karakteristik distribusi kemungkinan. Sebagai tambahan, satu konsumen mungkin datang dalam sekali waku atau dalam kelompok. Kelompok mungkin dalam ukuran konstan tau ukuran acak.

Model yang paling penting untuk kedatangan random adalah **Proses Kedatangan Poisson**

IV.2.2.6. Perilaku Antrian dan Disiplin Antrian.

Perilaku antrian menunjukkan bagaimana seorang yang berada pada sistem antrian tersebut berlaku atau bereaksi atas apa yang dialaminya dalam sistem antrian. Pertama, jika seseorang akan memasuki sebuah sistem antrian, dan dia melihat bahwa panjang antrian dalam sistem itu sangat banyak maka dia akan memutuskan untuk kembali dan tidak jadi memasuki sistem antrian. Kejadian ini disebut dengan nama ***balk.*** Kejadian lain adalah kejadian yang disebut ***renege*** yang menjelaskan bahwa seorang akan meninggalkan sistem antrian setelah dia memasuki sistem tersebut karena alasan panjangnya sistem antrian dan sistem itu berjalan sangat lambat menurutnya. Dan yang terakhir adalah kejadian jockey, yaitu berpindahnya seorang yang berada dalam sistem antrian ke sistem antrian lain yang dianggap lebih pendek atau memiliki tingkat pelayanan yang lebih pendek.

Disiplin Antrian adalag sebuah proses yang menunjukkan bagaimana aturan logis untuk memilih konsumen yang akan dilayani ketika stasiun pelayanan/server idle. Aturan pemilihan tersebut dapat berupa ***FIFO (First in First Serve) , LIFO (Last in First Serve), SIRO (Service in Random order), SPT (Shortest processing Time First), dan SPR (Service according to Priority)***. Dalam sebuah industri yang bersifat Job Shop, aturan antrian kadang berdasarkan pada waktu selsesai *(Due Date),* dan perkiraan waktu proses untuk sebuah *job* tertentu.

IV.2.3. Penyelesaian masalah antrian menggunakan bantuan software Microsoft Excel.

Untuk memudahkan kita bagaiamana prosedur dan cara menyelesaikan masalah antrian, yang kemudian kita dekati dengan penggunaan Microsoft Excell, ada beberapa prasyarat yang harus dimiliki oleh kita, yaitu:

1. Memahami konsep Simulasi (Table Simulation).
2. Memahami Konsep Probabilitas dan Distribusi Frekuensi.
3. Memahami konsep uji Distribusi dalam statistika.
4. Memahami cara perngoperasian Fungsi logika, matematika, dan Statistika dalam software Microsoft Excel.

Sebelumnya perlu diketahui, bahwa simulasi yang dapat diselesaikan dengan menggunakan Ms Excel sangat bergantung pada pembangkitan bilangan random yang dilakukan. Bilangan random yang telah dibangkitkan tersebut mewakili dari distribusi frekuensi/probabilitas dari kejadian yang ada.

Untuk langkah-langkah penyelesaiannya, secara detail akan diuraikan pada bagian selanjutnya.

IV.3. TUGAS PENDAHULUAN

Dari Studi Kasus yang terdapat di dalam BAB ini, maka buatlah rencana penyelsaian kasus tersebut dengan metoda simulasi (Langkah penyelesaian sebenarbnya akan dibahas pada praktikum) dengan ketentuan :

1. Mencakup langkah-langkah penyelesaiannya.
2. Kolom atau Tabel apa saja yang dibutuhkan untuk penyelesaian kasus tersbeut.
3. Perintah atau fungsi apa sajakh yang digunakan untuk penyelesaian kasus tersebut.
4. Model Diagram kasus tersebut.
5. Model statistik apakah yang digunakan dalam penyelesaian kasus tersebut.

IV.4. PROSEDUR PRAKTIKUM

1. Praktikan sudah menyelesaikan tugas pendahuluannya.
2. Pemberian materi dilakukan pada awal praktikum dengan metode diskusi serta pembahasan tugas pendahuluan.
3. Pembahasan tugas pendahuluan diarahkan pada langkah-langkah dan LKS yang ada pada modul.
4. Pembuatan program / pengisian LKS dipandu oleh asisten yang bersangkutan.
5. Pembacaan hasil/interpretasi dilakukan oleh praktikan dan dipandu oleh asisten.

IV.5. KASUS DAN CARA PENYELESAIAN

IV.5.1. Studi Kasus.

Berikut akan diberikan sebuah contoh aplikasi dari penggunakan Ms. Excell dalam penyelesaian masalah antrian sebuah stasiun pelayanan.

Kasus :

Sebuah sistem antrian yang terdiri dari satu buah stasiun pelayanan dengan distribusi pelayanan normal dan rata-rata waktu pelayanan 3.75 menit dan waktu antar kedatangan yang berdistribusi Poisson dengan waktu rata-rata antar kedatangan sebesar 3.4 Menit. Dari Obsevasi lapangan, yang dilakukan dengan 100 kali pengamatan secara random dan pada dua variabel tersebut diperoleh data sebagai berikut :

Waktu Pelayanan (Menit)

Waktu Pelayanan	Frekuensi terjadi
1	5
2	11
3	26
4	31
5	16
6	11
Total	**100**

Waktu Antar Kedatangan (menit)

Waktu antar kedatangan	Frekuensi Terjadi
1	5
2	30
3	25
4	20
5	10
6	5
7	3
8	2
Total	**100**

Tabel IV.2. Distribusi Frekuensi Studi Kasus Antrian

Kita ingin mengetahui, setelah jumlah pelanggan yang datang 50 orang, bagaimana keadaan sistem tersebut. Adapun hal-hal yang ingin diketahui meliputi :

1. Berapakah jumlah orang yang menunggu ?
2. Apakah server mengalami *idle*, Seberapa besar peluangnya ?
3. Berapakah peluang anda akan menunggu jika masuk dalam sistem antrian tersebut ?
4. Jika anda harus menunggu, berapa lama anda harus menjalaninya sebelum akhirnya dilayani ?
5. Berapa lama konsumen akan berada dalam sistem ?

IV.5.2. Prosedur Penyelesaian Masalah Antrian dengan Ms.Excel.

IV.5.2.1. Penetapan Variabel-Variabel yang Berubah.

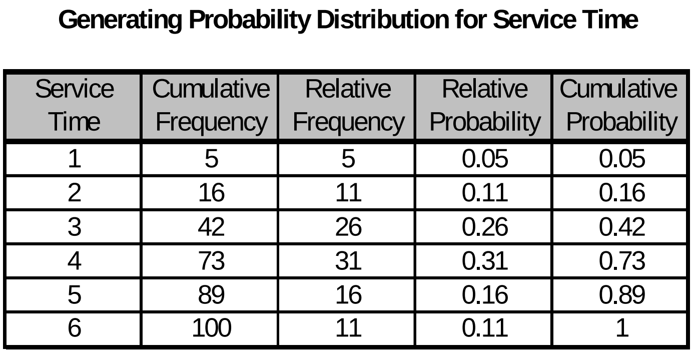
Dalam Simulasi, langkah pertama yang kita lakukan adalah mencoba mengetahui variabel-variabel apa saja yang terlibat dalam semuah sistem. Begitu pula dengan simulasi Antrian, kita berusaha menetapkan variabel apa saja yang terlibat dalam mekanisme antrian, setelah diadakan penelitian, misalnya kita mendapatkan bahwa mekanisme antrian bergantung pa variabel :

1. Waktu antar kedatangan.
2. Waktu pelayanan setiap server.
3. Jumlah server yang tersedia.
4. Waktu yang tersedia, dan
5. Jumlah kedatngan.

Setelah kita menentukan variabel-variabel yang terlibat, selanjutnya adalah menentukan dari variabel-variabel tersbut, mana yang selalu berubah-ubah dalam satuan waktu tertentu. Variabel itulah yang kemudian kita simulasikan sementara variabel yang lain kita asumsikan tetap. Misalnya kita menentukan bahwa hanya variabel **waktu antar kedatangan** dan **waktu pelayanan tiap server** yang selalu berubah dan akan kita simulasikan.

IV.5.2.2. Pengambilan data dari pengamatan.

Langkah selanjutnya adalah mengambil data dari pengamatan yang dilakukan. Kita akan berusaha melakukan pengamatan pada variabel yang dianggap selalu berubah yang dalam hal ini misalnya variabel waktu antar kedatangan dan waktu pelayanan tiap server. Kita kumpulkan data sampai dianggap cukup, lalu kita mengelompokkan data-data tersebut dengan kelas-kelas interval tertentu sebagaimana kita inginkan.



Generating Probability Distribution for Service Time

Service Time	Cumulative Frequency	Relative Frequency	Relative Probability	Cumulative Probability
1	5	5	0.05	0.05
2	16	11	0.11	0.16
3	42	26	0.26	0.42
4	73	31	0.31	0.73
5	89	16	0.16	0.89
6	100	11	0.11	1

Tabel IV.3. Distribusi probabilitas untuk waktu pelayanan

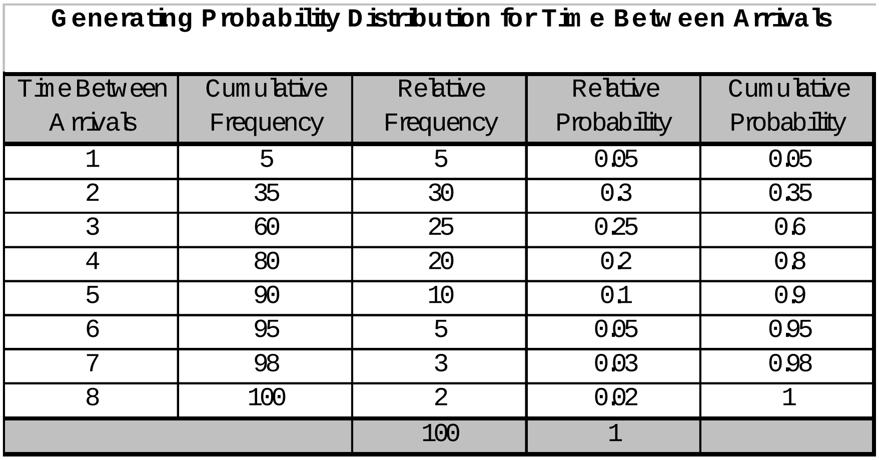
IV.5.2.3. Pembuatan Histogram dan Perkiraan Distribusi Frekuensi.

Dari interval kelas yang telah kita tentukan diatas, maka kita akan menghitung tally frekuensi dari masing-masing kelas tersebut, untuk kemudian kita buat Histogramnya..

Perlu diketahui, bahwa tujuan dibuatnya histogram dalam simulasi antrian disini, adalah agar kita dapat melihat distribuasi frekuensi yang terjadi dan berusaha memperkirakan sesuai dengan distribusi jenis apa kejadian yang kita amati. Frekuensi Variabel random yang kita amati bisa saja terlihat seperti distribusi *Poisson*, *Exponential*, *Normal*, *Binomial*, *Triangular*, *Weibull*, *Gamma*, atau bentuk distribusi frekuensi lainnya.

Dalam tahap ini, tugas kita yang penting adalah menentukan distribusi jenis apa yang sesuai dengan frekeunsi variabel yang kita amati.

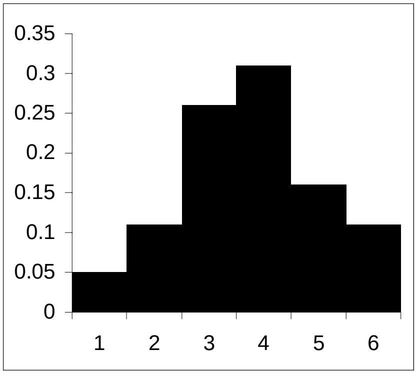
Kita akan membuat histogram dengan berbagai macam perintah yang ada pada Ms. Excel. Setelah dibuat histyogramnya, akan didapat sebagai berikut :

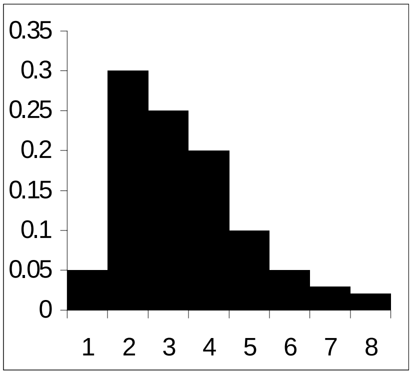


Generating Probability Distribution for Time Between Arrivals

Time Between Arrivals	Cumulative Frequency	Relative Frequency	Relative Probability	Cumulative Probability
1	5	5	0.05	0.05
2	35	30	0.3	0.35
3	60	25	0.25	0.6
4	80	20	0.2	0.8
5	90	10	0.1	0.9
6	95	5	0.05	0.95
7	98	3	0.03	0.98
8	100	2	0.02	1
		100	1	

Tabel IV.4. Distribusi probabilitas untuk waktu antar kedatangan





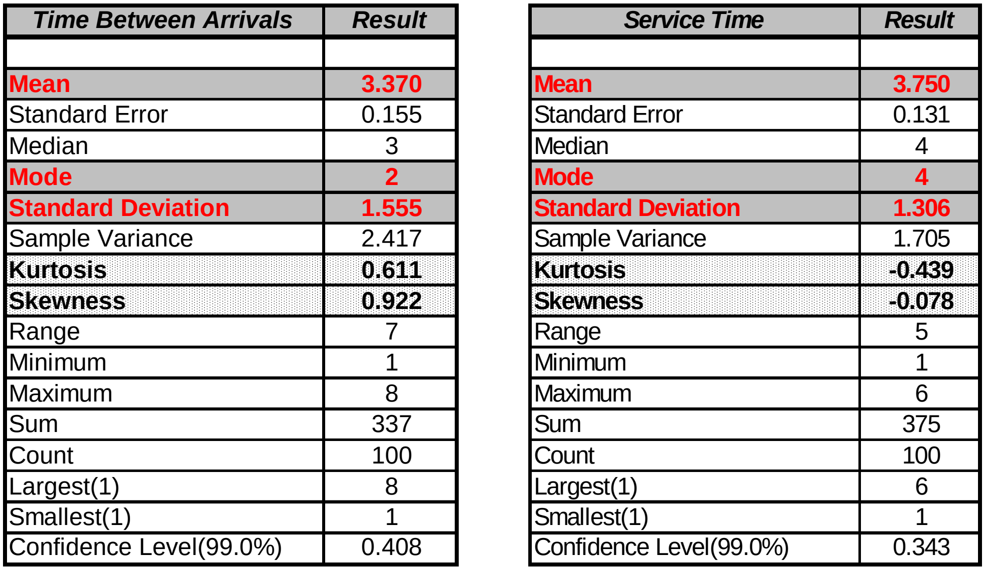
Gambar IV.6. Histogram Distribusi Frekuensi

<u>IV.5.2.4. Menentukan hasil Statistik Deskriptif dari pengamatan.</u>

Setelah kita menentukatelah kita menentukan dan mendapatkan bentuk distribusi yang sesuia dengan frekuensi variabel yang kita amati, maka langkah selanjutnya adalah menentukan hasil dari perhitungan statistik deskriptifnya n dan mendapatkan bentuk distribusi yang sesuia dengan frekuensi variabel yang kita amati, maka langkah selanjutnya adalah menentukan hasil dari perhitungan statistik deskriptifnya.

Statistik deskriptif adalah sebuah metoda statistik yang berusaha menjelaskan atau menggambarkan bagaimana strukur data pengamatan yang kita ambill. Statistik deskriptif lebih berhubungan dengan pengumpulan dan peringkasan data, serta penyajian hasil peringkasan tersebut. Data statistik – yang diperoleh dari pengamatan biasanya masih acak, “mentah” dan tidak terorganisir dengan baik. Untuk mempermudah dalam melakukan pengolahan, maka data-data tersebut harus diringkas dengan baik dan teratur, baik dalam bentuk tabel ataupun presentasi grafis, sebagai dasar untuk pengambilan keputusan (statistik inferensi).

Dalam penentuan statistik deskriptif ini kita berusaha menampilkan unsur-unsur yang penting seperti *mean, median, standar deviasi, kurtosis, sweakness*, ataupun elemen lain yang dianggap penting dalam sebuah presentasi tabel hingga secara kolektif kita dapat mengetahui karakteristik dari pengamatan yang kita lakukan.



Time Between Arrivals	***Result***
Mean	**3.370**
Standard Error	0.155
Median	3
Mode	**2**
Standard Deviation	**1.555**
Sample Variance	2.417
Kurtosis	**0.611**
Skewness	**0.922**
Range	7
Minimum	1
Maximum	8
Sum	337
Count	100
Largest(1)	8
Smallest(1)	1
Confidence Level(99.0%)	0.408

Service Time	***Result***
Mean	**3.750**
Standard Error	0.131
Median	4
Mode	**4**
Standard Deviation	**1.306**
Sample Variance	1.705
Kurtosis	**-0.439**
Skewness	**-0.078**
Range	5
Minimum	1
Maximum	6
Sum	375
Count	100
Largest(1)	6
Smallest(1)	1
Confidence Level(99.0%)	0.343

Tabel IV.5. Hasil Statistik Deskriptif

IV.5.2.5. Pengujian Distribusi (Goodnes of Fit)

Setelah kita memperkirakan Distribusi apa yang cocok dengan variabel hasil pengamatan, maka kita akan melakukan pengujian apakah distribusi yang gtelah kita estimasikan benar-benar sesuai dengan kenyataan yang ada. Hal sangat penting dilakukan karena fungsi yang ada pada distribusi tertentu ini akan kita gunakan pada pembangkitan nilai simulasi dan *random digit assignment*. Dalam hal ini kita bisa menggunakan berbagai macam cara yang sesuai dengan metoda statistika yang ada.Misalnya kita akan menggunakan uji ***Chi-Square Goodnes of Fit***, maka kita akan menuntukan :

H_0 : Frekuensi Relatif pengamatan sesuai dengan sebuah bentuk Distribusi frekuensi tertentu (Ditentukan)

H_1 : Frekuensi Relatif pengamatan tidak sesuai dengan sebuah bentuk distribusi frekuensi tertentu (yang telah ditentukan)

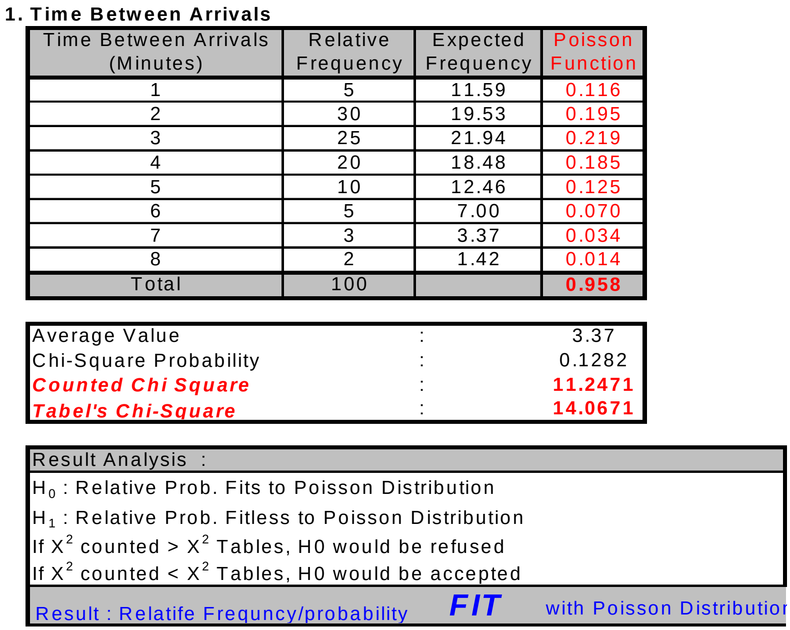
Setelah kita menentukan H_0 dan H_1 nya, maka langkah selanjutnya dengan rumus yang seusia dengan fungsi ***Chi-Square*** kita menentukan nilai $\mathbf{X^2}$ nya baik $\mathbf{X^2}$ hitung (dari pengamatan) maupun $\mathbf{X^2}$ Tabel. Kesimpulan yang didapat :

Jika $\mathbf{X^2}$ hitung > $\mathbf{X^2}$ tabel , maka H_0 ditolak .

Jika $\mathbf{X^2}$ hitung < $\mathbf{X^2}$ tabel , maka H_0 diterima.

Jika H_0 diterima, maka dapat disimpulkan bahwa frekuensi hasil pengamtan sesuai dengan sebuah bentuk distribusi frekuensi tertenu, dan jika ditolah maka sebaliknya.

Jika H_0 ditolak, maka kita harus mengulangi proses estimasi distribusi dan pengujian distribusi sampai ditemukan distribusi yang sesuai.



1. Time Between Arrivals

Time Between Arrivals (Minutes)	Relative Frequency	Expected Frequency	Poisson Function
1	5	11.59	0.116
2	30	19.53	0.195
3	25	21.94	0.219
4	20	18.48	0.185
5	10	12.46	0.125
6	5	7.00	0.070
7	3	3.37	0.034
8	2	1.42	0.014
Total	100		**0.958**

Average Value	:	3.37
Chi-Square Probability	:	0.1282
Counted Chi Square	:	**11.2471**
Tabel's Chi-Square	:	**14.0671**

Result Analysis :

H_0 : Relative Prob. Fits to Poisson Distribution

H_1 : Relative Prob. Fitless to Poisson Distribution

If X^2 counted > X^2 Tables, H0 would be refused

If X^2 counted < X^2 Tables, H0 would be accepted

Result : Relatife Frequncy/probability ***FIT*** with Poisson Distributior

Tabel IV.6. Hasil Uji Distribusi Waktu antar KEdatangan

2. Service Time

Service Time (Minutes)	Relative Frequency	Expected Frequency	Normal Function	Norm Cum Probability	Standardized Z
1	5	3.32	0.033	0.018	-2.106
2	11	12.44	0.124	0.090	-1.340
3	26	25.91	0.259	0.283	-0.574
4	31	30.00	0.300	0.576	0.191
5	16	19.32	0.193	0.831	0.957
6	11	6.92	0.069	0.958	1.723
Total	100				

Average Value	:	3.75
Standard Deviation	:	1.31
Chi-Square Probability	:	0.55
Counted Chi Square	**:**	**4.02**
Tabel's Chi-Square	**:**	**11.07**

Result Analysis :

H_0 : Relative Prob. Fits to Normal Distribution

H_1 : Relative Prob. Fitless to Normal Distribution

If X^2 counted > X^2 Tables, H0 would be refused

If X^2 counted < X^2 Tables, H0 would be accepted

Result : Relatife Frequncy/probability ***FIT*** with Normal Distribution

Tabel IV.7. Hasil Uji Distribusi Waktu Pelayanan

IV.5.2.6. Pembangkitan Nilai-Nilai Simulasi dan penentuan bilangan random

Setelah kita mengetahui secara kolektif karakteristik data yang kita amatai melalui pendekatan statistik deskriptif, maka masalah selanjutnya yang timbul adalah, bagaimana simulasi itu dapat dibangkitkan ?.

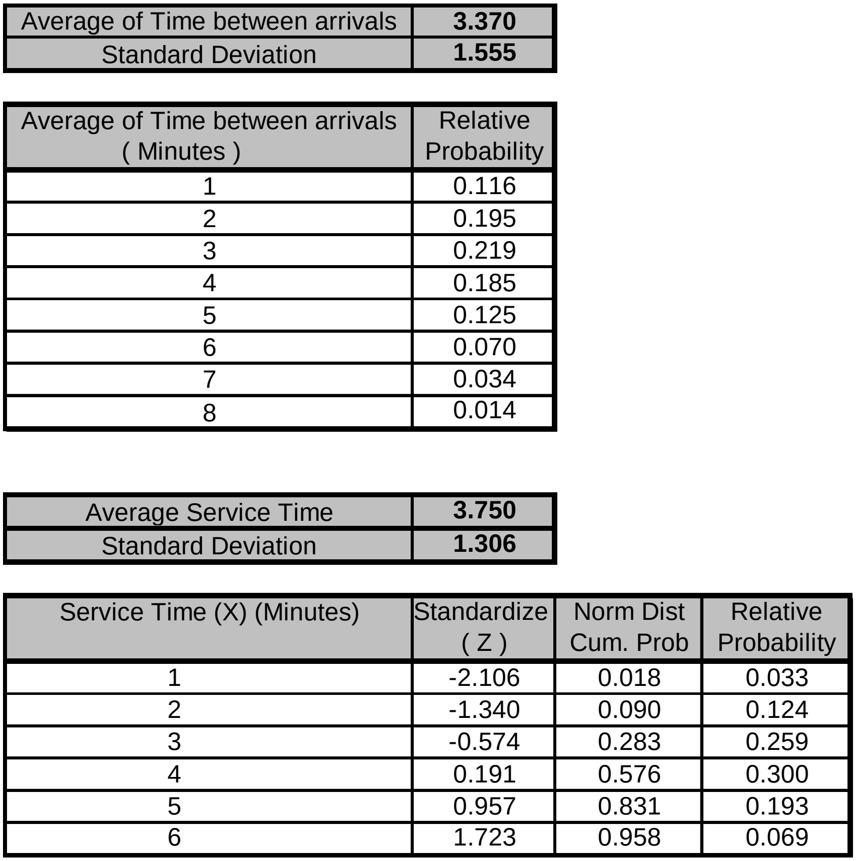
Diawal kita telah menentukan variabel yang terlibat dan selalu berubah dalam setiap keadaan waktu. Dalam kasus antrian ini misalnya diasumsikan bahwa variabel yang berubah adalah waktu antar kedatangan dan waktu pelayanan tiap server. Kita berusaha untuk memunculkan perubahan-perubahan yang terjadi pada setiap variabel dalam waktu selama simulasi itu berjalan. Perubahan – perubahan tersebut harus mengacu pada :

1. Mekanisme antrian yang sudah dipahami.
2. Distribusi frekuensi yang mewakili masing-masing variabel, dan
3. Bilangan random yang telah dibangkitkan.

Jadi, kita berusaha membangkitkan perubahan-perubahan yang terjadi pada setiap variabel dengan cara menentuakan bilangan random, dimana dalam suatu *range* tertentu, bilangan random tersebut mewakili nilai probabilitas dari kemungkinan keadaan pada setiap variabel. Dalam fase ini kita akan memetukan :

1. Probabilitas kemungkikan dari suatu keadaan variabel berdasarkan Fungsi distribusi yang sesuai.
2. Menentukan *range* bilangan random yang mewakili masing masing Probabilitas keadaan variabel.
3. Meng-*generate* bilangan random melalui Ms. Excel.

Mengenai penentuan probabilitas kejadian melalui fungsi distribusi, dapat dijelaskan misalnya kita akan menentukan probabilitas sebuah server dengan waktu pelayanan 3 menit dengan bentuk distribusi normal, maka yang kita lakukan adalah mencari probabilitas server tersebut menggunakan fungsi distribusi normal yang telah ditentukan.



Average of Time between arrivals	**3.370**
Standard Deviation	**1.555**

Average of Time between arrivals (Minutes)	Relative Probability
1	0.116
2	0.195
3	0.219
4	0.185
5	0.125
6	0.070
7	0.034
8	0.014

Average Service Time	**3.750**
Standard Deviation	**1.306**

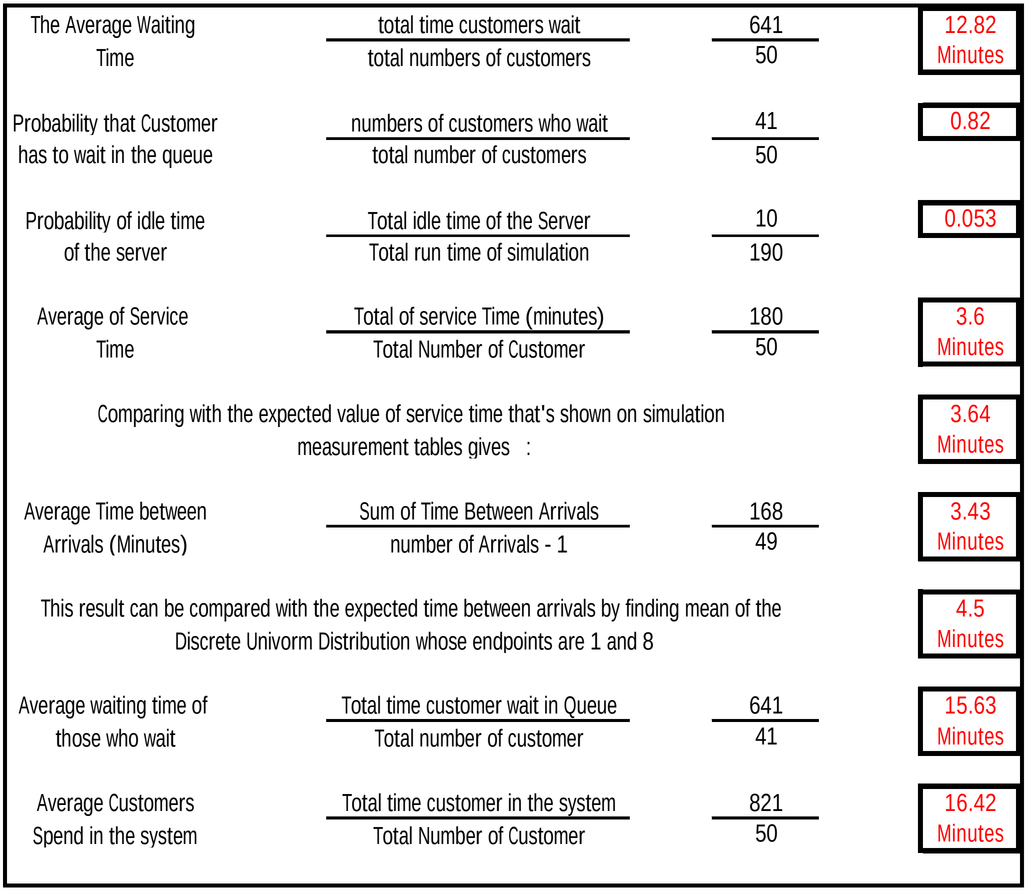
Service Time (X) (Minutes)	Standardize (Z)	Norm Dist Cum. Prob	Relative Probability
1	-2.106	0.018	0.033
2	-1.340	0.090	0.124
3	-0.574	0.283	0.259
4	0.191	0.576	0.300
5	0.957	0.831	0.193
6	1.723	0.958	0.069

Tabel IV.8. Pembangkitan Probabilitas Untuk Simulasi

IV.5.2.7. Menjalankan Simulasi Tabel.

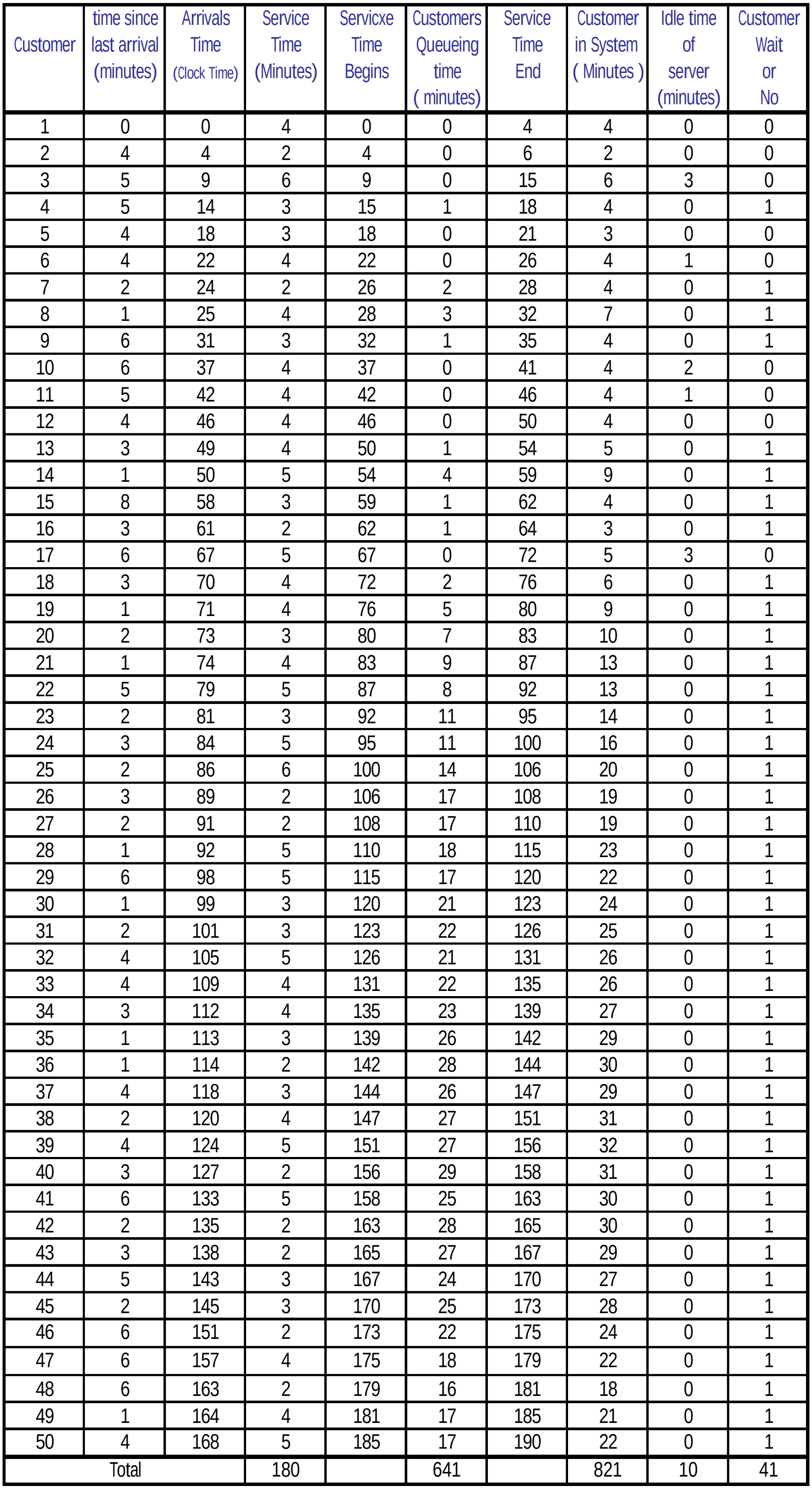
Setelah semua elemen dalam simulasi kita persiapkan, maka langkah terakhir adalah menjalankan simulasi, dalam Ms Excell, simulasi yang dilakukan biasanya disajikan dalam bentuk tabel, hingga sering disebut sebagai simulasi tabel.

Sesuai dengan prinsip simulasi, bahwa hal itu harus dilakukan secara berulang dan eksperimentatif, maka untuk replikasinya dalam aplikasi Ms. Excell dapat dilakukan dengan menekan tombol ***Delete*.** Setiap kita menekan tombol *delete* pada *keyboard* komputer, maka bilangan trandom akan berubah dimana masing maisng bilangan random terebut mengacu pada sebuah kejadian dari variabel yang berdistribusi tertentu. Simulasi dapat dihentikan sampai jumlah replikasi dianggap mencukupi, untuk setelah itu dianalisis hasilnya.



The Average Waiting Time	total time customers wait total numbers of customers	641 50	12.82 Minutes
Probability that Customer has to wait in the queue	numbers of customers who wait total number of customers	41 50	0.82
Probability of idle time of the server	Total idle time of the Server Total run time of simulation	10 190	0.053
Average of Service Time	Total of service Time (minutes) Total Number of Customer	180 50	3.6 Minutes
Comparing with the expected value of service time that's shown on simulation measurement tables gives :			3.64 Minutes
Average Time between Arrivals (Minutes)	Sum of Time Between Arrivals number of Arrivals - 1	168 49	3.43 Minutes
This result can be compared with the expected time between arrivals by finding mean of the Discrete Univorm Distribution whose endpoints are 1 and 8			4.5 Minutes
Average waiting time of those who wait	Total time customer wait in Queue Total number of customer	641 41	15.63 Minutes
Average Customers Spend in the system	Total time customer in the system Total Number of Customer	821 50	16.42 Minutes

Gambar IV.7. Hasil Simulasi Antrian



Customer	time since last arrival (minutes)	Arrivals Time (Clock Time)	Service Time (Minutes)	Servicxe Time Begins	Customers Queueing time (minutes)	Service Time End	Customer in System (Minutes)	Idle time of server (minutes)	Customer Wait or No
1	0	0	4	0	0	4	4	0	0
2	4	4	2	4	0	6	2	0	0
3	5	9	6	9	0	15	6	3	0
4	5	14	3	15	1	18	4	0	1
5	4	18	3	18	0	21	3	0	0
6	4	22	4	22	0	26	4	1	0
7	2	24	2	26	2	28	4	0	1
8	1	25	4	28	3	32	7	0	1
9	6	31	3	32	1	35	4	0	1
10	6	37	4	37	0	41	4	2	0
11	5	42	4	42	0	46	4	1	0
12	4	46	4	46	0	50	4	0	0
13	3	49	4	50	1	54	5	0	1
14	1	50	5	54	4	59	9	0	1
15	8	58	3	59	1	62	4	0	1
16	3	61	2	62	1	64	3	0	1
17	6	67	5	67	0	72	5	3	0
18	3	70	4	72	2	76	6	0	1
19	1	71	4	76	5	80	9	0	1
20	2	73	3	80	7	83	10	0	1
21	1	74	4	83	9	87	13	0	1
22	5	79	5	87	8	92	13	0	1
23	2	81	3	92	11	95	14	0	1
24	3	84	5	95	11	100	16	0	1
25	2	86	6	100	14	106	20	0	1
26	3	89	2	106	17	108	19	0	1
27	2	91	2	108	17	110	19	0	1
28	1	92	5	110	18	115	23	0	1
29	6	98	5	115	17	120	22	0	1
30	1	99	3	120	21	123	24	0	1
31	2	101	3	123	22	126	25	0	1
32	4	105	5	126	21	131	26	0	1
33	4	109	4	131	22	135	26	0	1
34	3	112	4	135	23	139	27	0	1
35	1	113	3	139	26	142	29	0	1
36	1	114	2	142	28	144	30	0	1
37	4	118	3	144	26	147	29	0	1
38	2	120	4	147	27	151	31	0	1
39	4	124	5	151	27	156	32	0	1
40	3	127	2	156	29	158	31	0	1
41	6	133	5	158	25	163	30	0	1
42	2	135	2	163	28	165	30	0	1
43	3	138	2	165	27	167	29	0	1
44	5	143	3	167	24	170	27	0	1
45	2	145	3	170	25	173	28	0	1
46	6	151	2	173	22	175	24	0	1
47	6	157	4	175	18	179	22	0	1
48	6	163	2	179	16	181	18	0	1
49	1	164	4	181	17	185	21	0	1
50	4	168	5	185	17	190	22	0	1
	Total		180		641		821	10	41

Gambar IV.8. Simulasi Tabel untuk Antrian

IV.6. ANALISIS HASIL SIMULASI

Dari hasil yang didapat, ternyata dapat diambil beberapa kesimpulan, seperti :

1. Jumlah pelanggan yang harus menunggu dalam antrian adalah 41 orang, artinya hanya sembilan orang yang langsung dilayani dari percoban yang dilakukan pada 50 orang.
2. Waktu menunggu rata-rata dari pelanggan sebelum ia dilayani adalah 12.82 menit.
3. Kemungkinan pelanggan akan menunggu jika memasuki sistem antrian tersbeut adalah 0.82, artinya hampir bisa dipastikan setiap pelanggan yang memasuki sistem antrian harus menunggu selama 12.82 menit.
4. Server sangat sibuk, dibuktikan bahwa hanya 0.053 kemungkinan server untuk menganggur.
5. Rata-rata waktu pelanggan berada dalam sistem antrian adalah 16.42 menit, artinya bahawa dari mulai ia masuk ke dalam sistem antrian sampai ia keluar memerlukan waktu selama kurang lebih 16.42 menit.

Dari hasil tersebut dapat dianalisis mengapa terjadi antrian yang sangat panjang. Pertama – tama kita mengetahui bahwa waktu rata-rata pelayanan lebih lama dari waktu rata-rata antar kedatangan, sehingga dapat dipastikan akan adanya antrian dan semakin lama tentu antrian akan semakin panjang. Berikutnya jika kita lihat bahwa waktu hasil simulasi berbeda dengan perhitungan statistik maupun pendekatan kuantitatif lainnya :

Metoda	Waktu Antar Kedatangan	Waktu Pelayanan
Statistik Deskriptif	**3.37 menit**	**3.75 menit**
Simulasi	**3.43 menit**	**3.6 menit**
Nilai Harapan		**3.64 menit**
Distribusi Uniform	**4.5 menit**	

Tabel IV.9. Perbandingan Hasil Simulasi

Dari tabel diatas terlihat bahwa pada Hasil simulasi untuk waktu antar kedatangan memiliki nilai paling rendah dari keempat metode yang ada, dilain pihak untuk waktu pelayanan hasil simulasi memiliki nilai yang paling tinggi diantara metoda kuantitatif yang dilakukan. Dengan demikian ada kecenderungan baha peritungan panjang antrian hasil simulasi berbeda dengan perhitungan yang menggunakan pendekatan lain. Dan kecenderungan itu terlihat bahwa panjang antrian hasil simulasi akan lebih besar dari perhitungan panjang antrian metoda kuantitatif lain. Begitu pula halnya dengan kemungkinan orang harus menunggu dan waktu yang dihabiskan pelangga dalam sistem. Namun yang perlu diperhatikan, bahwa kesimpulan seperti itu tidaklah bersifak general, namun sangat kasuistis, sehingga kesimpulan seperti tersebut diatas tidak boleh diaplikasikan pada sistem yang lain, tanpa melakukan eksperimen dan pengamatan pada hasil simulasinya.

Permasalahannya sekarang adalah, mengapa hal itu bisa berbeda ?. Sebelum menjawab hal tersebut perlu kita ingat kembali, bahwa metode simulasi bukanlah sebuah metode optimasi yang hanya memiliki satu titik penyelesaian yang dianggap optimal. Metode simulasi adalah sebuah pendekatan sistem, dimana kita berussha merepresentasikan sistem selengkap mungkin. Tidaklah seperti pada pendekatan optimasi yang kadang harus disertai asumsi yang menyebabkan pemodelan sistemnya kuarang pas dan menitikberatkan pada hasil, artinya dengan metoda kuantitaif tersebut kita hanya bisa melihat hasil tanpa mengetahui bagaimana proses dari perubahan sistem tersebut berlangsung. Metoda simulasi mencoba menjelaskan bagaimana proses suatu sistem berlangsung dan berubah dari waktu ke waktu dimana kita dapat mengamati perubahan yang terjadi. Hal itu menyebabkan hasil yang diperoleh dari metode simulasi merupakan pendekatan dari proses berjalannya sistem, atau kenyataan yang ada. Dan hal itu tidak bisa dilakukan hanya dengan satu kali percobaan. Replikasi yang banyak mutlak diperlukan, karena dalam kenyataannya-pun hampir mustahil bahwa kejadian sistem selalu sama dari waktu ke waktu. Dengan demikian hasil dari sebuah proses simualsi tidaklah tunggal seperti pada metoda kuantitatif lainnya, melainkan ada banyak titik yang masing-masing mewakili dari output sistem yang selalu berubah tiap waktunya.

Nah, sekarang kita kembali ke kasus diatas, dari hasil yang terdapat di tabel terlihat bahwa hasil dari berbagai macam pendekatan ternyata berbeda, artinya walaupun dari perhitungannya, metoda-metoda yang lain dapat dipertangungjawabkan nilai kebenarannya, namun masih memiliki resiko yang cukup besar jika didasarkan pengambilan keputusan pada sistem terutama yang berskala besar. Memang betul, dalam kasus ini perbedaannya masih dalam tingkat beberapa menit saja, namun anda bisa membayangkan jika hal itu akan diimplementasikan pada sebuah sistem yang sangat besar dimana perdetiknya menyangkut dana atau sumder daya yang cukup besar, maka hal tersebut akan mengakibatkan kerugian yang tidak sedikit pula.

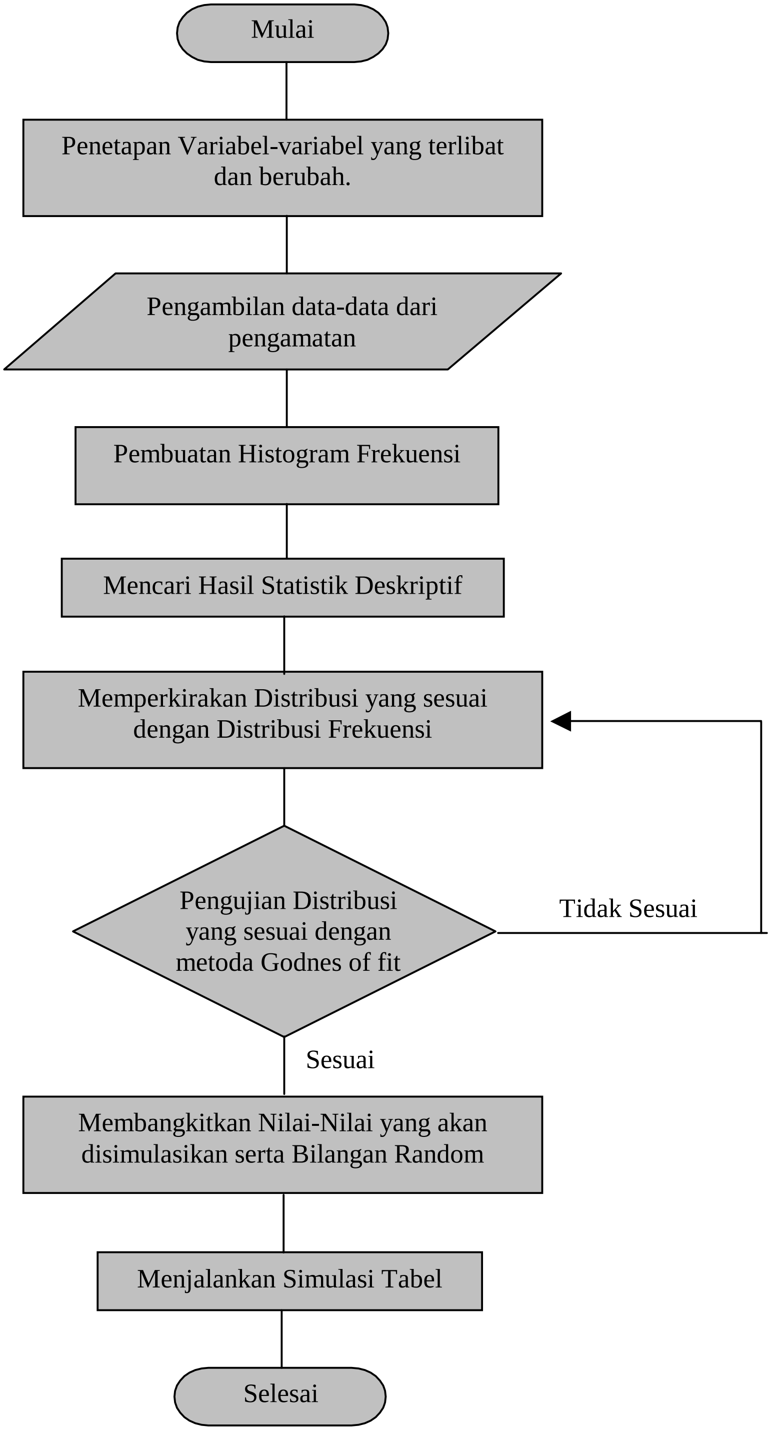
Lalu bagaimana simulasi menjawab permasalahan diatas ?. Sebagaimana kita sudah jelaskan tadi, bahwa simulasi darus dilakukan dengan ekperimen-eksperimen yang disertai replikasi yang cukup.

Pertama, kita tentutakan dulu apa yang mempengaruhi panjang antrian, ataupun apa yang menyebabkan permasalahan tersebut timbul. Misalnya dari analisis didapat bahwa yang menjadi penyebab adalah waktu pelayanan yang terlalu lama dan waktu antar kedatangan yang tinggi. Nah, dari sini apa yang hendak kita lakukan ? Kita tahu, bahwa untuk variabel waktu antar kedatangan cukup sulit bagi kita untuk merubahnya, karena hal tersebut datangnya dari bagian eksternal sistem yang disebut *Calling Population.* Sehingga kita hanya bisa merubah variabel waktu pelayanan saja. Lalu ada masalah yang timbul, yaitu sampai tingkat berapa waktu pelayanan harus diubah tanpa harus menyebabkan waktu pelayanan saja. Lalu ada masalah yang timbul, yaitu sampai tingkat berapa waktu pelayanan harus diubah tanpa harus menyebabkan server idle ?. Sekali lagi, jangan dilakukan dengan pendekatan kuantitatif-matematis, jangan menerapkan rumus-rumus yang ada dalam kaidah optimasim, untuk mendapatkan waktu pelyanan yang ideal. Namun gunakanlah eksperimen-eksperimen. Caranya Anda dapat menurunkan waktu pelayanan dalam fase deskriptif statistik, lalu lakukan sebuah simulasi, lihat hasilnya. Jika masih kurang dan masih terdapat antrian, kembali lagi pada fase statistik deskriptif, ubah lagi waktu pelayanannya dengan angka yang lebih kecil, lakukan lagi simulasi, dan lihat lagi hasilnya. Hal ini dilakukan terus menerus sampai mendapatkan hasil yang diinginkan. Dan perlu diingatkan kembali dalam melakukan tiap eksperimen harus dibarengi dengan replikasi yang cukup.

Setelah tahap tersebut selesai, kita akan mengetahui pada tingkat pelayanan berapa sistem antrian akan optimal. Namun Simulasi bukanlah optimasi, dan simulasi bukanlah alat pemecahan yang sempurna, simulasi digunakan untuk merepresentasikan atau memodelkan suatu sistem nyata yang ada. Sehingga walaupun kita mendapatkan suatu tingkat pelayanan yang optimal, masih diperlukan pendekatan lain untuk dapat menjelaskan bagaimana kita bisa mencapai tingkat waktu pelayanan optimal tersebut.

IV.7. FLOW CHART PRAKTIKUM

Berikut ini adalah Diagram Alur prosedur penyelesaian Simulasi Antrian dengan MS. Excel.:


Mulai
Penetapan Variabel-variabel yang terlibat dan berubah.
Pengambilan data-data dari pengamatan
Pembuatan Histogram Frekuensi
Mencari Hasil Statistik Deskriptif
Memperkirakan Distribusi yang sesuai dengan Distribusi Frekuensi
Pengujian Distribusi yang sesuai dengan metoda Godnes of fit
Tidak Sesuai
Sesuai
Membangkitkan Nilai-Nilai yang akan disimulasikan serta Bilangan Random
Menjalankan Simulasi Tabel
Selesai

Gambar IV.9. Flow Chart Penyelesaian Masalah

Introduction

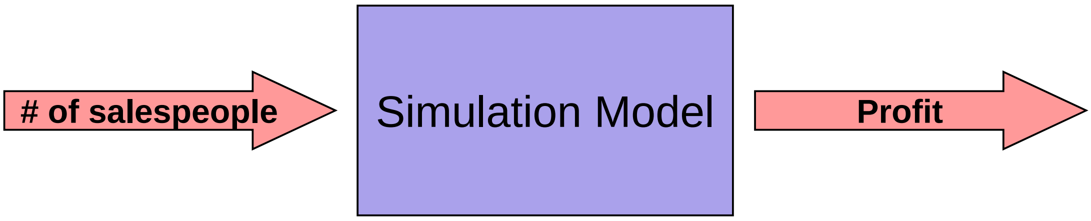
Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

What is simulation?

- *Experiments* with a *computer model* of a *real system*
- *Systematic* variation of *input* data to study the effect on *output* data
- Allows investigation of *consequences* of different decisions



Dynamic, Stochastic Simulation

- **Stochastic** implies that one can take *uncertainty* and *risk* into account

- **Dynamic** implies the ability to follow processes in detail over time

- Widely used:
  - In large corporations
  - In medium size companies
  - In governments
  - In public service activities

Wide Use in Business

- Production, inventory and purchase planning
- Manning and capacity decisions
- Forecasting of cash flows
- Forecasting the market for new products
- Costing

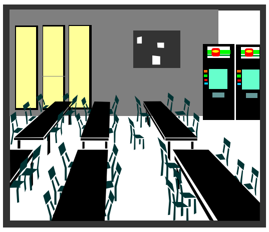
Simulation in Education

- From high school through colleges and universities
- In mathematics courses for studying statistics and probability
- In marketing courses and in production planning courses
- In hospital administration in health care

Benefits to Students

- Modeling *existing* systems
- Modeling proposals for *improving* an existing system
- Practice in *data acquisition*
- Training in elementary *statistical analysis* and *experimental design*

School Cafeteria Simulation…
What *random* input data would need to be collected?
What *control* data would be collected?



GPSS

- **G**eneral **P**urpose **S**imulation **S**ystem
- Originated with IBM
- Advantages of GPSS
  - Programs are *similar* to real problems
  - Programs can be developed *quickly*
  - *Graphic* system facilitates understanding
  - Automatic production of relevant statistics
- micro-GPSS
  - Developed at the Stockholm School of Economics
  - used in more than 40 countries

WebGPSS

- A modern version of micro-GPSS
- Client component provides a GUI for developing simulation models graphically
- Server component provides the ability to run simulation models over the Web
- Contains on-line lessons providing student guidance
- Provides many help texts

WebGPSS

Lesson 1

Simulation Practical Lab Work Activities using WebGPSS

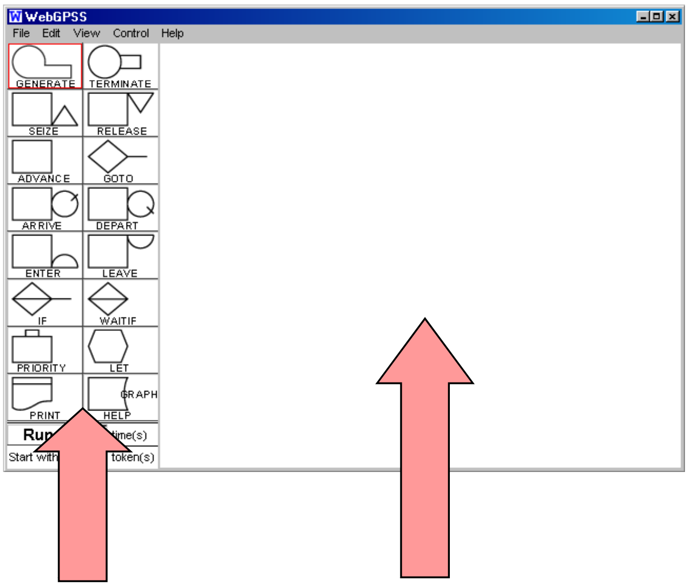
Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

The Lesson 1 Simulation Model

- The system is a *turnstile* at a booth that sells *tokens*.
- Customers arrive exactly *18 minutes* apart to buy tokens, each customer buying exactly one token.
- There are a total of *50 tokens at the start* of the day.
- When all 50 tokens are taken, the booth must *close down* for the day.

The WebGPSS Window

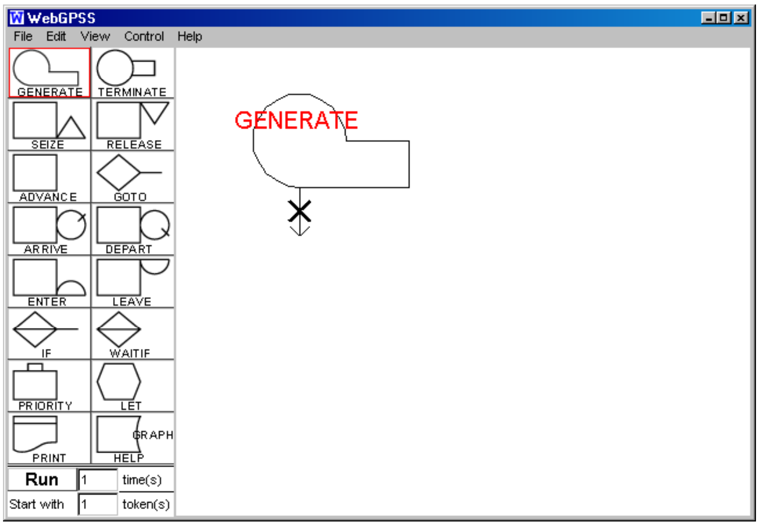


Symbol Menu

Block diagram layout area

- Programs are built in the layout area by clicking on symbols in the symbol menu.

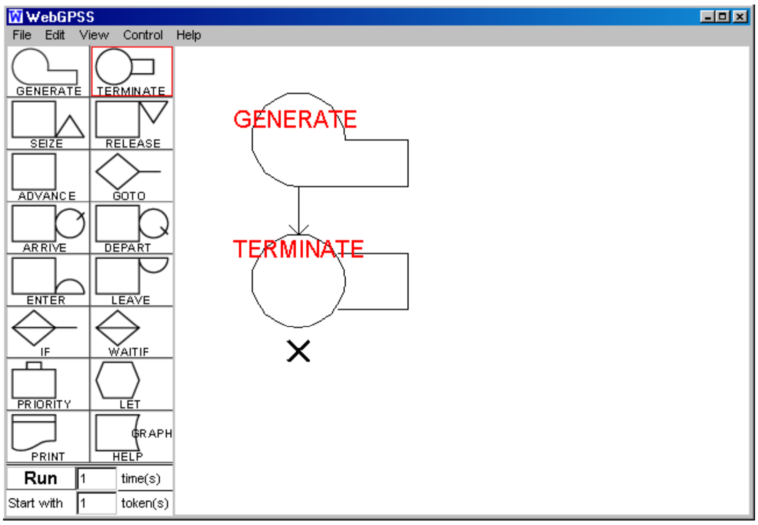
The GENERATE Block



- Clicking on the GENERATE symbol in the symbol menu creates a copy of it in the block diagram layout area.
- The X indicates where the next block will be positioned.

- GENERATE is the first block used in every GPSS program.
- It is used to introduce customers (known as transactions in GPSS) into our simulation.

The TERMINATE Block

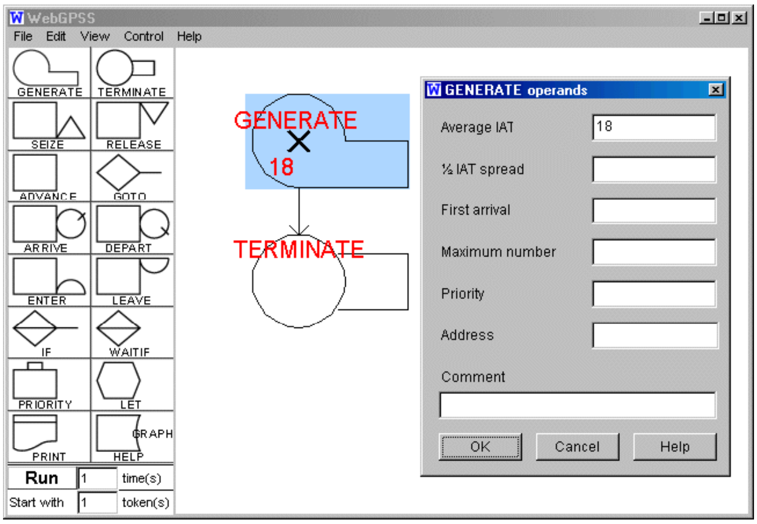


Run 1 time(s)
Start with 1 token(s)

- These two blocks make up what is called a *segment* in GPSS.
- The arrow shows the movement of customers from the GENERATE to the TERMINATAE blocks.

- The TERMINATE block gets customers out of the simulation. (We sometime say in GPSS that the transaction is destroyed.)

GENERATE Operands



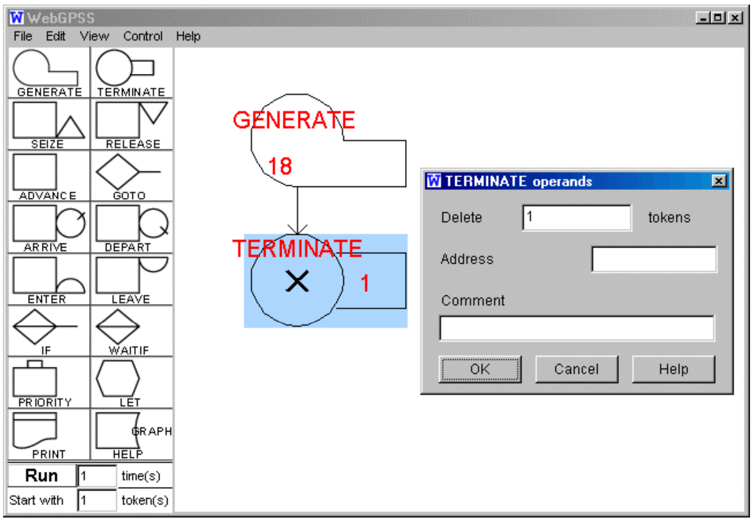
Run 1 time(s)
Start with 1 token(s)

Average IAT 18
½ IAT spread
First arrival
Maximum number
Priority
Address
Comment

- Double-clicking on the GENERATE block opens the GENERATE operands window.
- The block is marked with a light blue color.

- We specify 18 as the average IAT (InterArrival Time) for customers.
- Generate allows for five operands, but we only use the first of these (Average IAT) in this simulation. (GPSS refers to a block's first operand as the A operand.)

TERMINATE Operands



Delete 1 tokens

Address

Comment

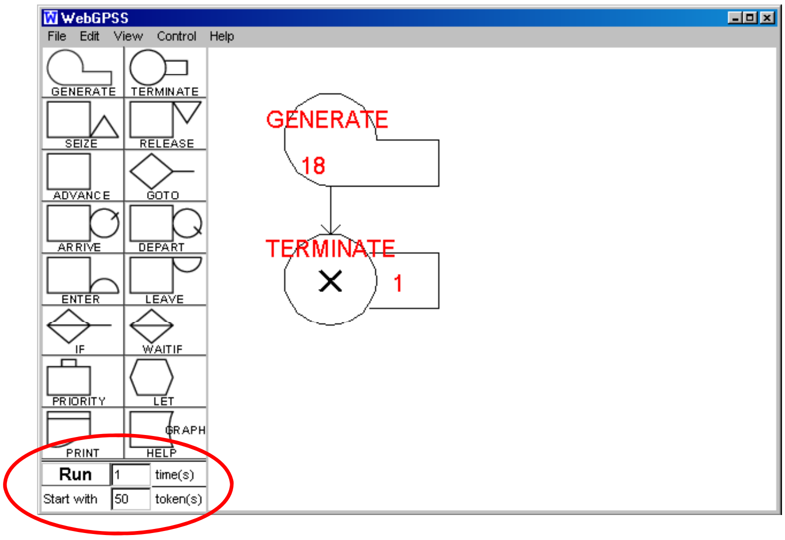
Run 1 time(s)

Start with 1 token(s)

- The TERMINATE block has only one operand, the number of tokens to be taken away.
- This is said to be the A operand for the TERMINATE block.

- Since each customer buys only one token, we type a 1 in the *Delete* operand.
- This completes the block diagram for our first program.

Running the Simulation Program



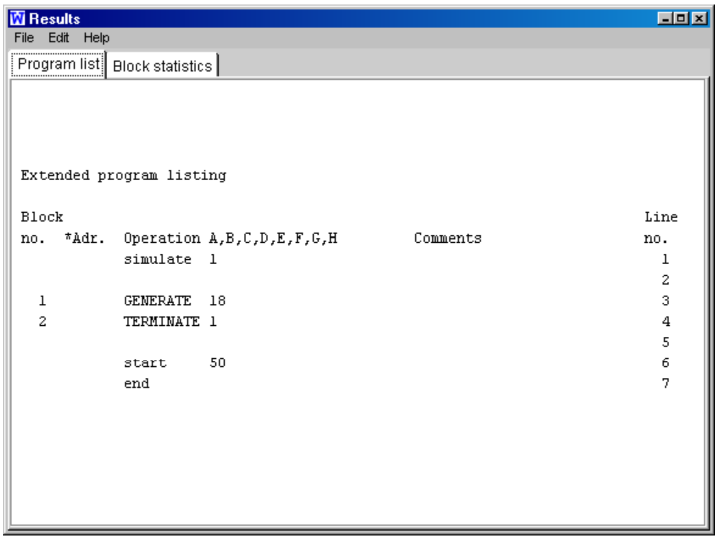
Run 1 time(s)

Start with 50 token(s)

- Since we want to run our program once, we leave the default value of 1 for the number of *times* to *Run*.
- Since we want 50 customers to go through the turnstile, we set the *Start with* box to 50 tokens.

- Clicking on the Run button then runs our simulation program.
- After a short wait, the Results window appears, with two tabs for statistics, the *program list* and *block statistics*.

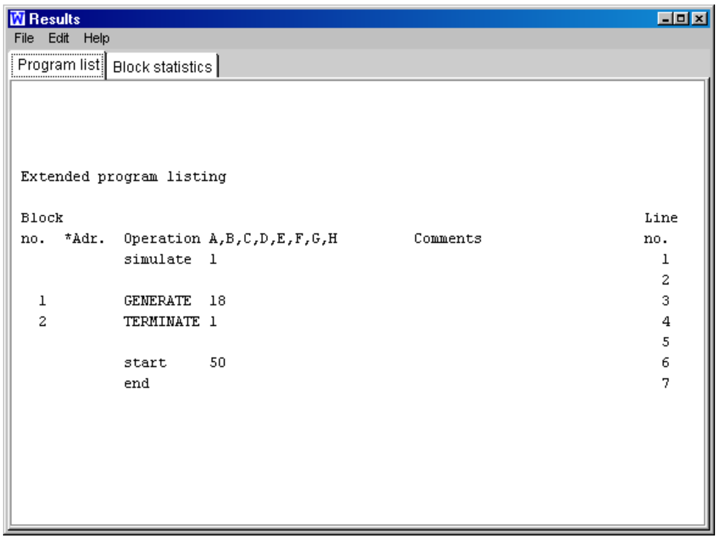
The Results Window – Program List



```
Extended program listing

Block                                                      Line
no.  *Adr.  Operation A,B,C,D,E,F,G,H        Comments      no.
            simulate  1                                     1
                                                            2
  1         GENERATE  18                                    3
  2         TERMINATE 1                                     4
                                                            5
            start     50                                    6
            end                                             7
```

- The Extended program listing contains some additional features added by the GPSS system, including the headings, the *block numbers*, and *line numbers*.
- The GPSS system has also added three control statements, required in every GPSS program:
  - SIMULATE
  - START
  - END

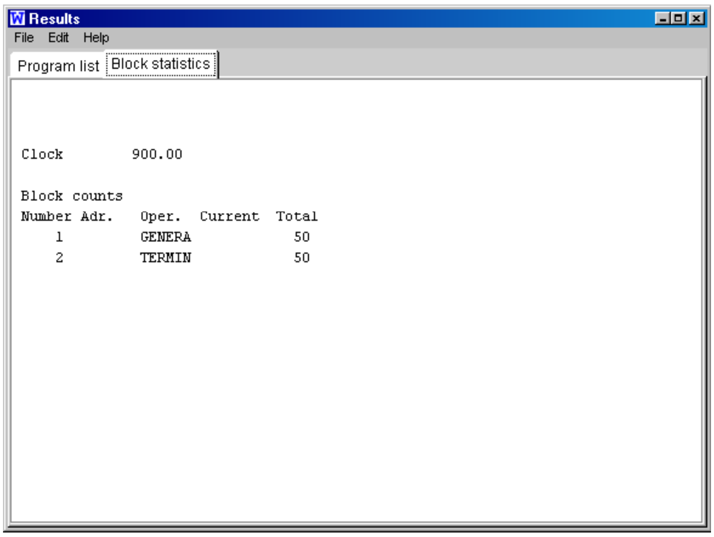
The Required Control Statements



```
Extended program listing

Block                                                      Line
no.  *Adr.  Operation A,B,C,D,E,F,G,H        Comments      no.
            simulate  1                                     1
                                                            2
  1         GENERATE  18                                    3
  2         TERMINATE 1                                     4
                                                            5
            start     50                                    6
            end                                             7
```

- **SIMULATE**
  - Always the first control statement in a GPSS program
  - The A operand is the number of times to run the program
- **END**
  - Required in every GPSS program
  - Marks the end of the GPSS program
- **START**
  - its A operand is the number of tokens available at the start of the simulation (sometimes referred to as the initial value of the *termination counter*).
  - Determines when the simulation stops, that is, when all tokens have been removed by transactions entering TERMINATE.

The Results Window - Block Statistics

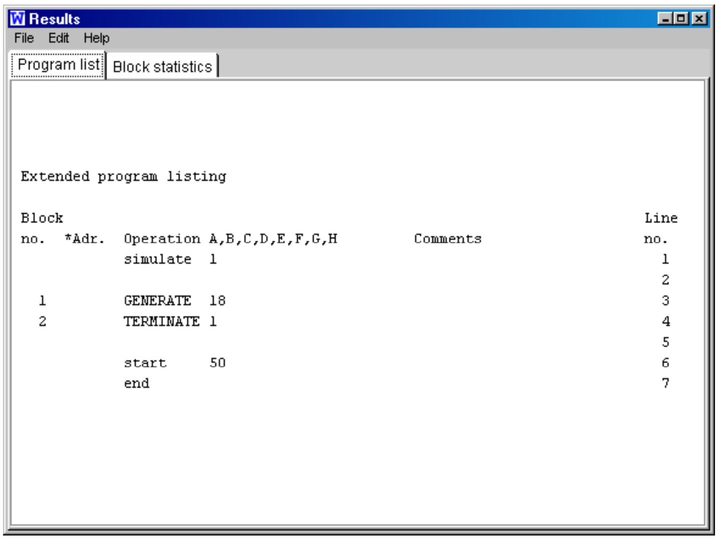


```
Clock        900.00

Block counts
Number Adr.    Oper.  Current  Total
    1          GENERA            50
    2          TERMIN            50
```

- **The Simulation Clock**
  - The simulation stopped at time 900 minutes, when the 50th customer went through the turnstile.

- **Total Block Counts**
  - We see that 50 customers had gone through the GENERATE block.
  - These same 50 customers had gone through the TERMINATE block.

Discussion Questions



```
Extended program listing

Block                                                           Line
no.   *Adr.   Operation A,B,C,D,E,F,G,H        Comments         no.
              simulate  1                                        1
                                                                 2
  1           GENERATE  18                                       3
  2           TERMINATE 1                                        4
                                                                 5
              start     50                                       6
              end                                                7
```

- *Question #1* What change in this program would make customers arrive every 24.6 minutes?
- *Question #2* What change in this program would make customers arrive every 10 hours?
- *Question #3* What change in this program would result in the arrival and exit of 100 customers?
- *Question #4* What would happen if the A operand of the terminate block was changed from a 1 to a 2?

Lesson 2

Simulation Practical Lab Work Activities using WebGPSS

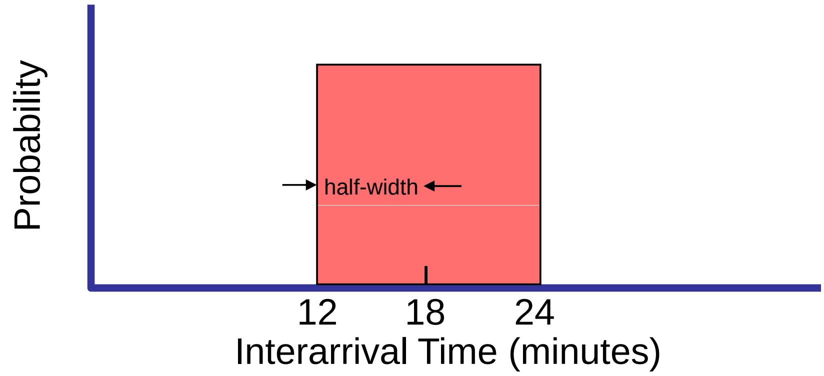
Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

The Lesson 2 Simulation Model

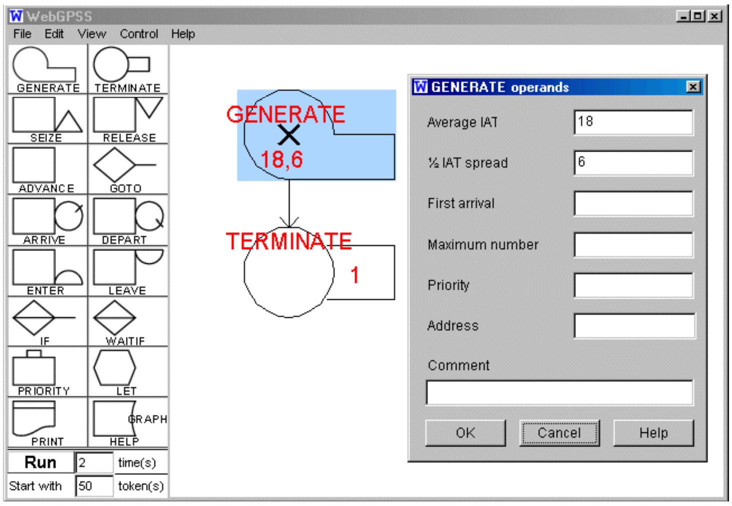
- The system is a *turnstile* at a booth that sells *tokens*.
- Customers arrive **randomly**, between 12 and 24 minutes apart, uniformly distributed on this interval.
- There are a total of *50 tokens at the start* of the day.
- When all 50 tokens are taken, the booth must *close down* for the day.

The Uniform Distribution



- The average IAT is still 18 minutes.
- Customers arrive 18 ± 6 minutes apart.
- The “half-spread” or “half-width” is 6 minutes, i.e., half of the total width of 24-12 = 12 minutes.

Random Interarrivals



Run 2 time(s)

Start with 50 token(s)

Average IAT 18

½ IAT spread 6

First arrival

Maximum number

Priority

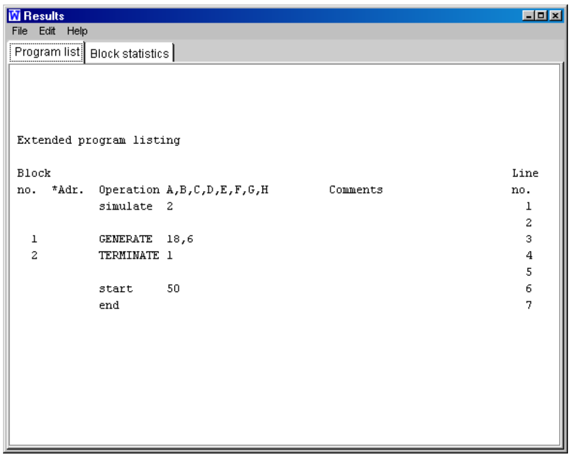
Address

Comment

- One change in the program of the previous lesson is to enter 6 in the “½ IAT spread” field.
- Two operands then show up in the block diagram, separated by commas.

- With randomness in this model, we will run the program two times, by adjusting the Run button field to 2 times.
- Every run will use a different sequence of random numbers, and will therefore provide different results.

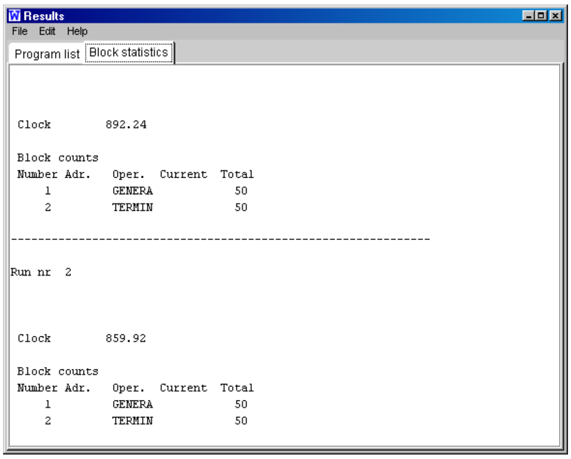
Results of Random IATs



```
Extended program listing

Block                                                        Line
no.   *Adr.  Operation A,B,C,D,E,F,G,H        Comments        no.
             simulate  2                                       1
                                                               2
  1          GENERATE  18,6                                    3
  2          TERMINATE 1                                       4
                                                               5
             start     50                                      6
             end                                               7
```

- Note that the Program list now has SIMULATE 2. The A operand 2 is the number of times to run.
- Note also that there are two operands in the GENERATE block:
  - *The A operand is the average IAT.*
  - *The B operand is the ½ IAT spread, or half-width.*
- The default value of the B operand of GENERATE is 0.

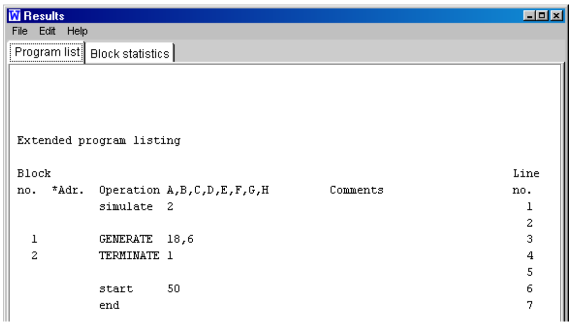
Results of Random IATs



```
 Clock        892.24

 Block counts
 Number Adr.   Oper.  Current  Total
     1         GENERA            50
     2         TERMIN            50

--------------------------------------------------------------

Run nr  2


 Clock        859.92

 Block counts
 Number Adr.   Oper.  Current  Total
     1         GENERA            50
     2         TERMIN            50
```

- Each simulation finished at a different time:
  - Run number 1 finished at 892.24 minutes.
  - Run number 2 finished at 859.92 minutes.
- Everyone who runs this program will get the same results, because GPSS produces ***pseudo-random*** numbers determined by a computer program.

Discussion Questions



```
Extended program listing

Block                                                    Line
no.  *Adr.  Operation A,B,C,D,E,F,G,H        Comments    no.
            simulate  2                                   1
                                                          2
  1         GENERATE  18,6                                3
  2         TERMINATE 1                                   4
                                                          5
            start     50                                  6
            end                                           7
```

- Customers arrive at an exhibition booth on average every 10 minutes, but sometimes only 5 minutes apart, and sometimes as much as a quarter-hour apart, with IATs uniformly distributed.
- Each customer takes a brochure, and the booth begins each morning with 80 brochures.

- *Question #1* How would you modify the GENERATE block for this program?
- *Question #2* How would you modify the TERMINATE block?
- *Question #3* How would you modify the START control statement?
- *Question #4* What modification would be necessary if each customer took **two** brochures?
- *Question #5* What if we started the day with only 79 brochures? How many brochures would the last customer get?

Lesson 3

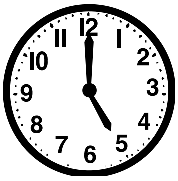
Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

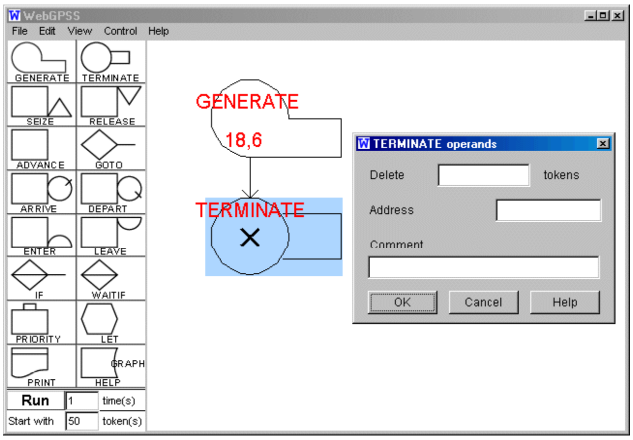
UIN Sunan Kalijaga Yogyakarta

The Lesson 3 Simulation Model

- The system is a *turnstile* at a booth that sells *tokens*.
- Customers arrive **randomly**, between 12 and 24 minutes apart, uniformly distributed on this interval.
- The turnstile will close and the **simulation will stop at a specific time**, namely after 8 hours.



TERMINATE with Default Operand



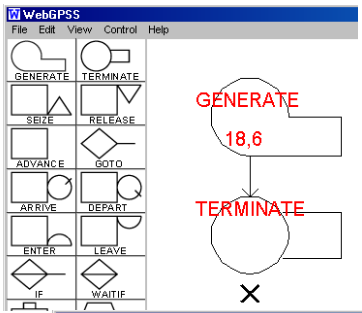
- We don’t want the customers to determine when the simulation stops, so we no longer have them remove tokens.
- We remove the 1 in the Delete operand of the terminate block—the default is 0.
- This adjusted terminate removes the customers from the system but does not affect the value of the termination counter.

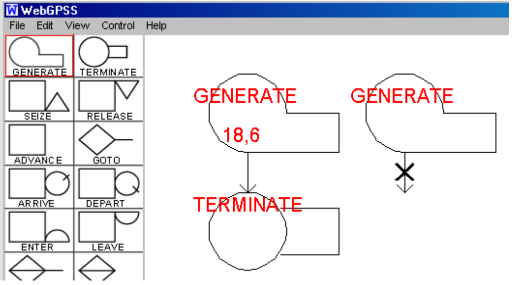
Segments

- We need to introduce a second segment into this program that will stop the simulation at 8 hours.

A segment is a sequence of blocks, started with GENERATE and usually ended with a TERMINATE block.

- We can think of this new segment as having a janitor who arrives at 8 hours to close the turnstile.
- To introduce the GENERATE for a new segment, we need to click just below the TERMINATE for the customer segment, introducing an X.
- We then click on GENERATE in the symbol menu, and the GENERATE block appears in the layout area, to the right of the customer segment’s GENERATE.

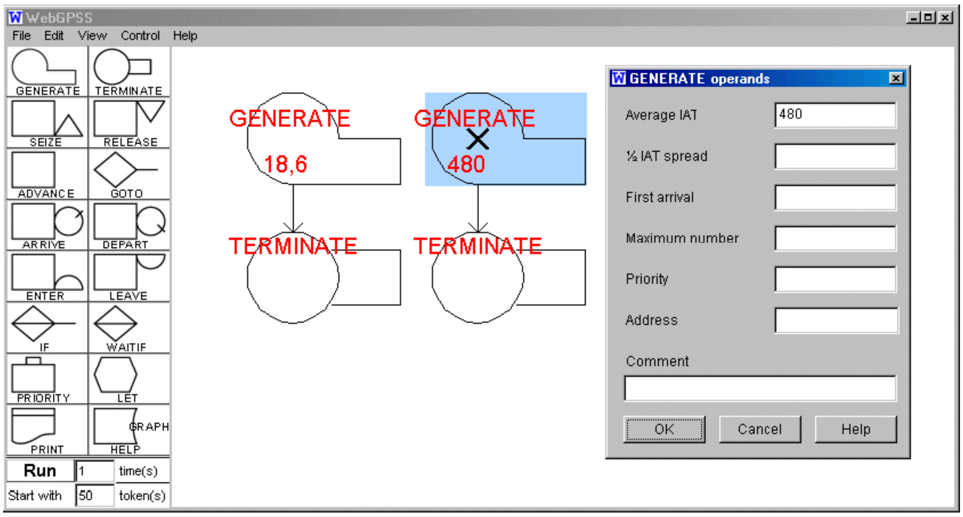




The Stop Segment

3

WebGPSS



Run 1 time(s)

Start with 50 token(s)

Average IAT: 480

½ IAT spread:

First arrival:

Maximum number:

Priority:

Address:

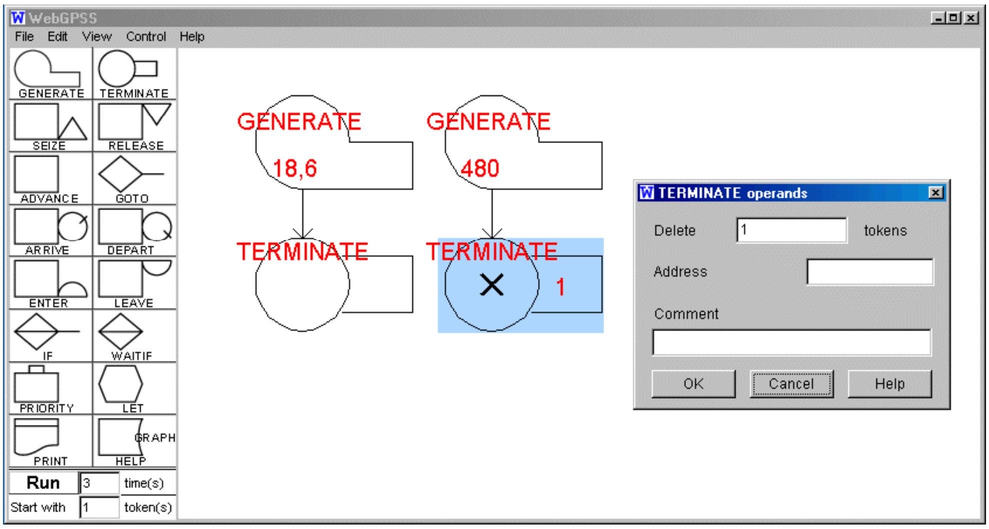
Comment:

- We introduce a TERMINATE block after the GENERATE block in the stop segment.
- Since we must use the same time unit throughout a GPSS model, we have the janitor arrive at time 480 minutes (8 hours). 60 min/hr x 8 hours = 480 minutes.

The Stop Segment

3

WebGPSS



Run 3 time(s)

Start with 1 token(s)

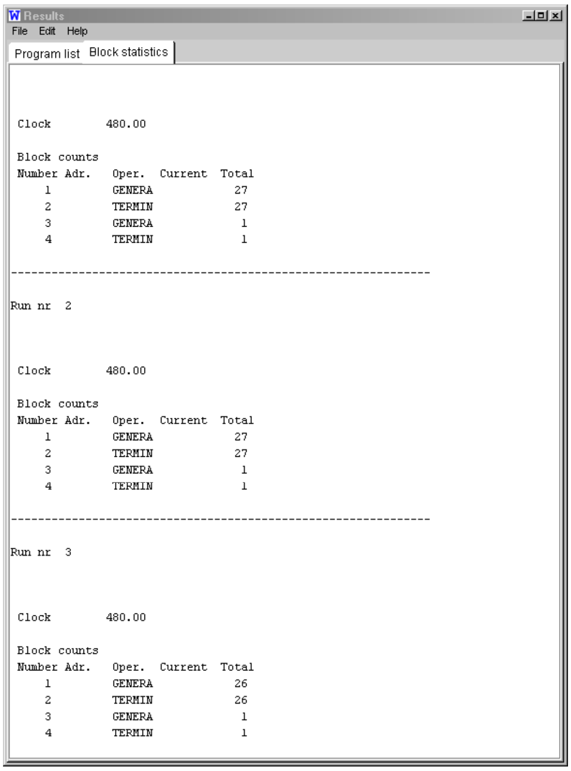
Delete 1 tokens

Address:

Comment:

- We want the janitor to delete one token, so we set its operand accordingly in the TERMINATE block.
- We "Start with" 1 token, so that when the janitor removes this token, the simulation will stop.
- We shall run this program 3 times.

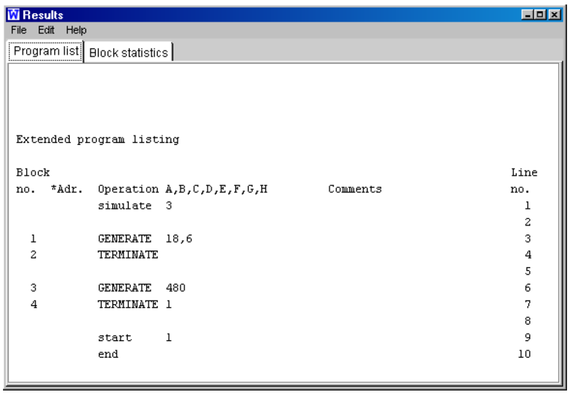
Results with the Stop Segment



```
 Clock        480.00

 Block counts
 Number Adr.   Oper.  Current  Total
     1         GENERA            27
     2         TERMIN            27
     3         GENERA             1
     4         TERMIN             1

--------------------------------------------------------------

Run nr  2


 Clock        480.00

 Block counts
 Number Adr.   Oper.  Current  Total
     1         GENERA            27
     2         TERMIN            27
     3         GENERA             1
     4         TERMIN             1

--------------------------------------------------------------

Run nr  3


 Clock        480.00

 Block counts
 Number Adr.   Oper.  Current  Total
     1         GENERA            26
     2         TERMIN            26
     3         GENERA             1
     4         TERMIN             1
```

- Notice the number of customers arriving in each of the three runs.
- How is it possible for the third run to have one fewer customers arrive?

Discussion Questions



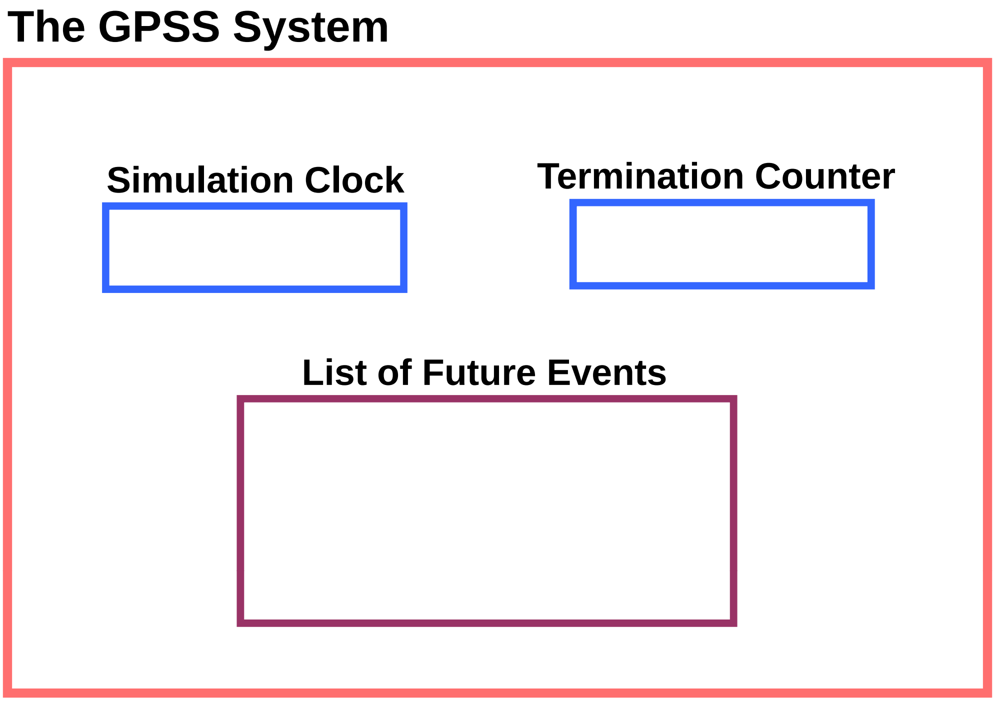
```
Extended program listing

Block                                                        Line
no.  *Adr.  Operation A,B,C,D,E,F,G,H        Comments         no.
            simulate  3                                         1
                                                                2
  1         GENERATE  18,6                                      3
  2         TERMINATE                                           4
                                                                5
  3         GENERATE  480                                       6
  4         TERMINATE 1                                         7
                                                                8
            start     1                                         9
            end                                                10
```

- *How would you modify this program so that customer IATs are 12 ± 6 minutes, uniformly distributed, and so that it will stop:*

  When the 40th customer walks through the turnstile **OR**

  When 480 minutes have elapsed, **whichever happens FIRST?**

A First View of the GPSS System



The GPSS System

Simulation Clock

Termination Counter

List of Future Events

GPSS Execution & Event Scheduling

1. The GPSS processor reads the SIMULATE and START A statements, and sets the TERMINATION COUNTER to the value specified in the A operand of the START statement.
2. The very first arrival for each segment’s GENERATE block is scheduled and placed on the LIST OF FUTURE EVENTS.
3. The list of future events is searched for the most immediate event.
4. The simulation CLOCK is moved to the event time.
5. If the event is a GENERATE event:
   - A new transaction is brought into the system.
   - The event is removed from the future events list.
   - The next arrival is scheduled and added to the list.
6. The transaction is brought forward as far as possible through the blocks.

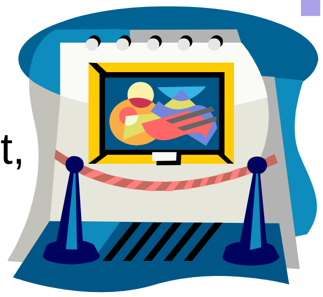
Lesson 4

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

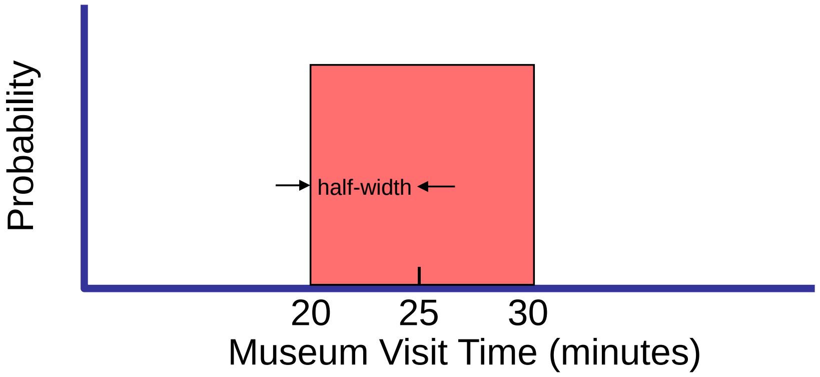
UIN Sunan Kalijaga Yogyakarta

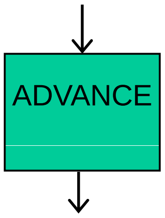
The Lesson 4 Simulation Model



- The system is a *museum*.
- Visitors arrive **randomly**, between 12 and 24 minutes apart, uniformly distributed on this interval.
- Visitors spend random times between 20 and 30 minutes in the museum, uniformly distributed on this interval.
- We do not limit the number of visitors who can be in the same museum at the same time.
- The museum will close and the **simulation will stop** after 8 hours of operation.

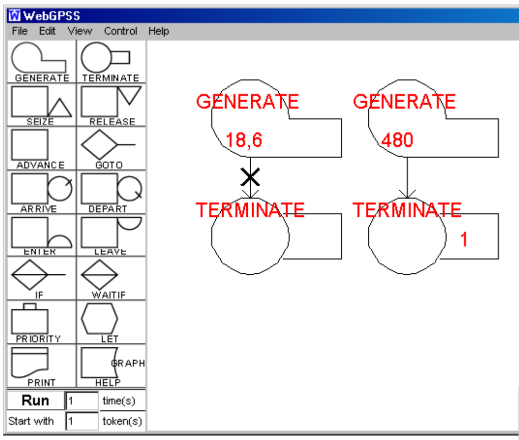
The ADVANCE Block



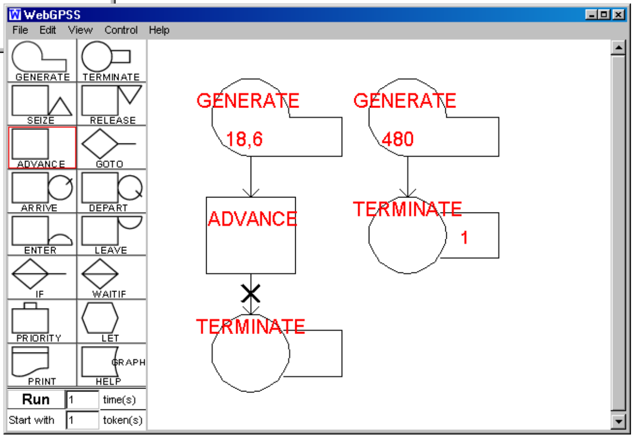


- The ADVANCE block holds a transaction (visitor, in this model) for a specified amount of time.
- Here, we want the average to be 25 minutes, with a half-width of 5 minutes, giving a random time between 20 and 30 minutes.

Inserting the ADVANCE Block

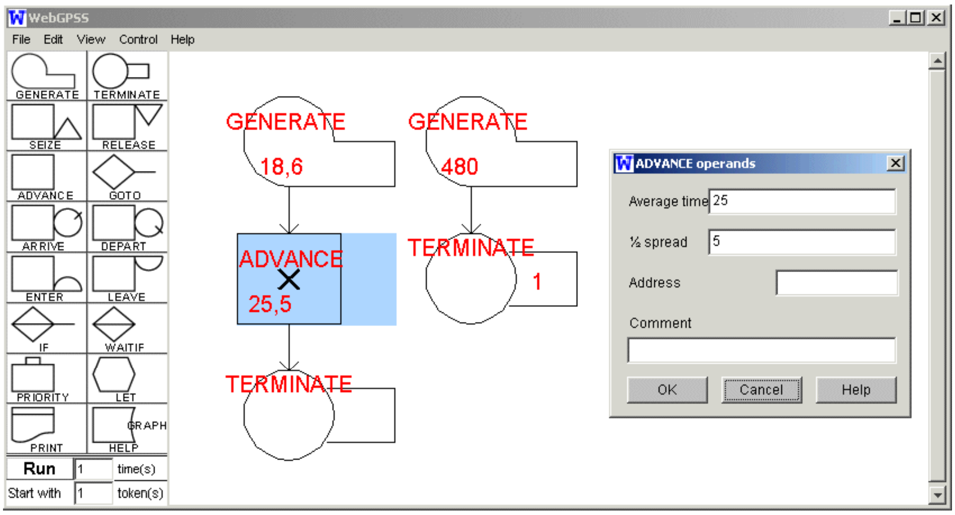


- To insert the ADVANCE block, we first click on the line connecting GENERATE and TERMINATE in the customer segment.
- The X indicates where this block will be inserted in our model.



- We then click on the ADVANCE symbol in the symbol menu.
- GPSS inserts the ADVANCE block between the GENERATE and TERMINATE blocks.

The ADVANCE Block Operands

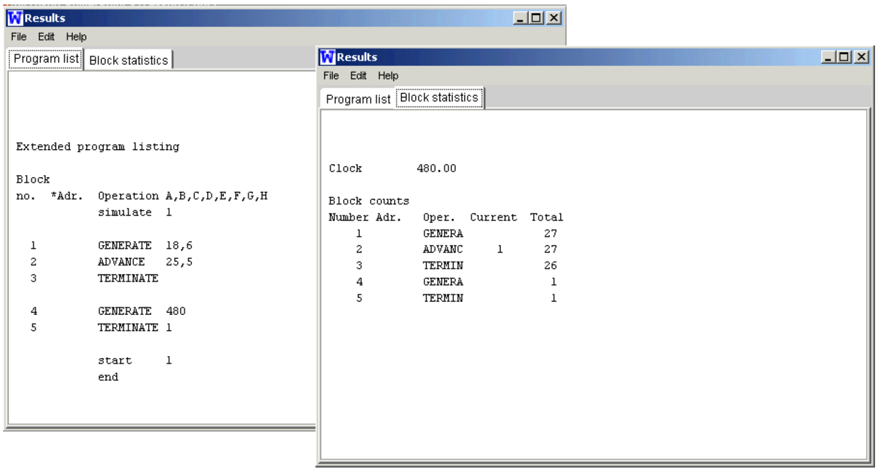


Run 1 time(s)
Start with 1 token(s)

Average time 25
½ spread 5
Address
Comment

- Double-clicking the ADVANCE block opens the ADVANCE Operands window.
- We key and enter 25 for the value of the *Average time* operand.
- We key and enter 5 for the value of the *½ spread* operand.
- These operands, separated by a comma, then appear in the ADVANCE block. (These are sometimes called the A and B operands in GPSS.)

Museum Simulation Results

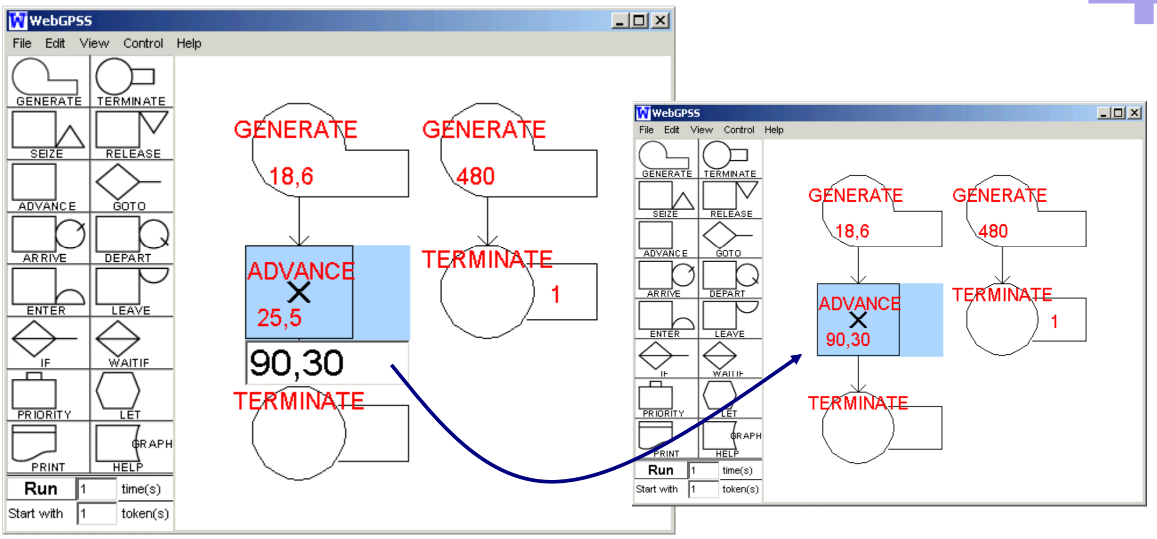


```
Extended program listing

Block
no.  *Adr.  Operation A,B,C,D,E,F,G,H
            simulate  1

  1         GENERATE  18,6
  2         ADVANCE   25,5
  3         TERMINATE

  4         GENERATE  480
  5         TERMINATE 1

            start     1
            end
```

```
Clock        480.00

Block counts
Number Adr.    Oper.  Current  Total
     1         GENERA             27
     2         ADVANC      1      27
     3         TERMIN             26
     4         GENERA              1
     5         TERMIN              1
```

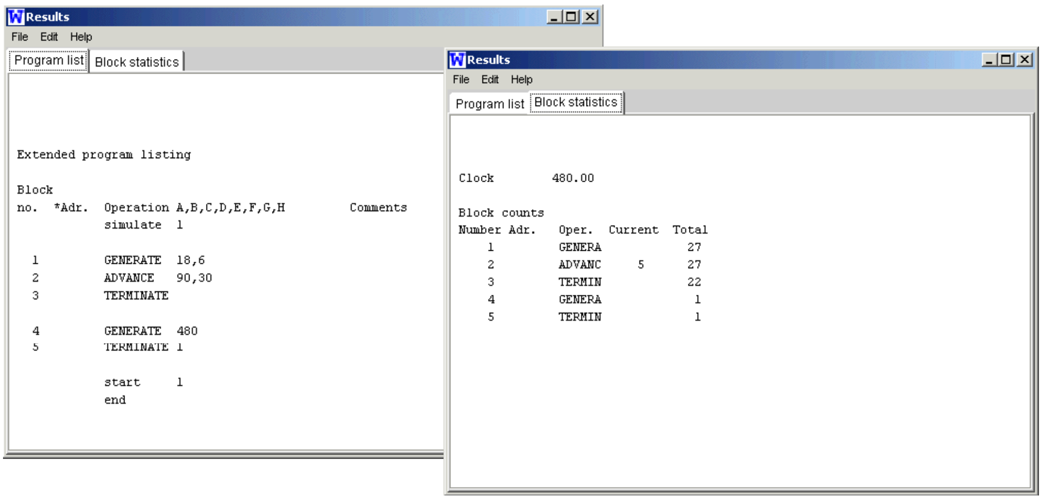
- New to the block statistics is data in the column labeled *Current*.
- **Question #1** What does the 1 in the ADVANCE block's *current* column tell us regarding this museum simulation?
- **Question #2** Why is the *total* count for the TERMINATE block number 3 only 26?
- **Question #3** How would you modify this program if visitors stayed in the museum between 1 and 2 hours, randomly and uniformly distributed on this interval?

Museum Visitors Stay 1 to 2 Hours



- Once you know the operand rules for a block, you can save time by NOT opening the Operands window for that block.
- Simply **single**-click on the block, and it will be marked in blue.
- Then, when you start typing your operands, they will appear in a text box below the blue-marked block.
- After keying in the operands this way, pressing the Enter key will place the new values inside of the block.

Results of 1 to 2 Hour Museum Stay



```
Extended program listing

Block
no.  *Adr.  Operation A,B,C,D,E,F,G,H        Comments
            simulate  1

  1         GENERATE  18,6
  2         ADVANCE   90,30
  3         TERMINATE

  4         GENERATE  480
  5         TERMINATE 1

            start     1
            end
```

```
Clock        480.00

Block counts
Number Adr.   Oper.   Current  Total
    1         GENERA             27
    2         ADVANC      5      27
    3         TERMIN             22
    4         GENERA              1
    5         TERMIN              1
```

- There is no specific limit on how many museum visitors can be in the ADVANCE block at the same time.
- **Question** Why is it NOT surprising to see the higher current count of 5 in the ADVANCE block than we saw when the visitors stayed between 20 and 30 minutes?

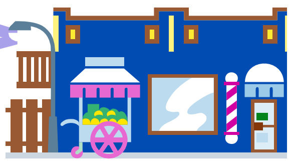
Lesson 5

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

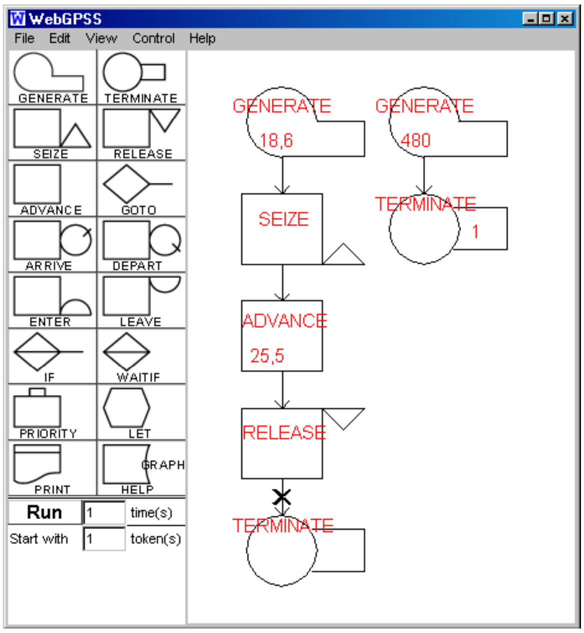
UIN Sunan Kalijaga Yogyakarta

The Lesson 5 Simulation Model



- The system is a *small barbershop,* with a single barber, whose name is Joe.
- Customers arrive **randomly**, between 12 and 24 minutes apart, uniformly distributed on this interval.
- The time for a haircut varies between 20 and 30 minutes, uniformly distributed on this interval.
- Joe can serve only one customer at a time, so a waiting line will form if Joe cannot keep up with the customer arrivals.
- The barbershop will close and the simulation will stop after 8 hours of operation.

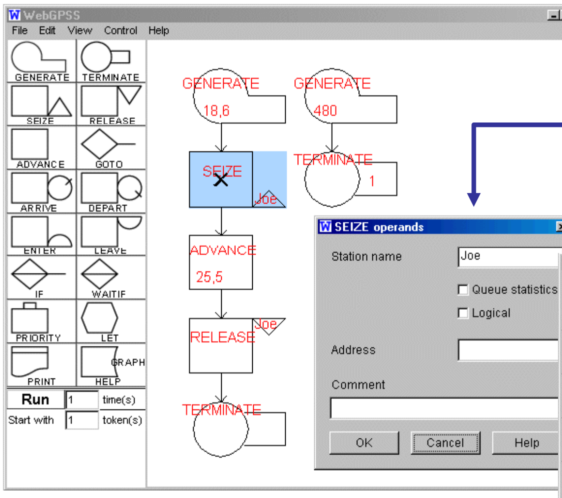
The SEIZE and RELEASE Blocks



- Begin with the museum model and insert two new blocks:
  - the SEIZE block
  - the RELEASE block
- The SEIZE block allows a customer to try to get service from a service station that can serve only one customer at a time.
- If the station is busy, then a waiting line (queue) will build up.
- The RELEASE block causes the customer to free the station so that other customers can be served.

- SEIZE and RELEASE are mirror images of each other.
- They allow only ONE customer at a time in the ADVANCE block.

SEIZE and RELEASE Block Operands



SEIZE operands

Station name: Joe

□ Queue statistics

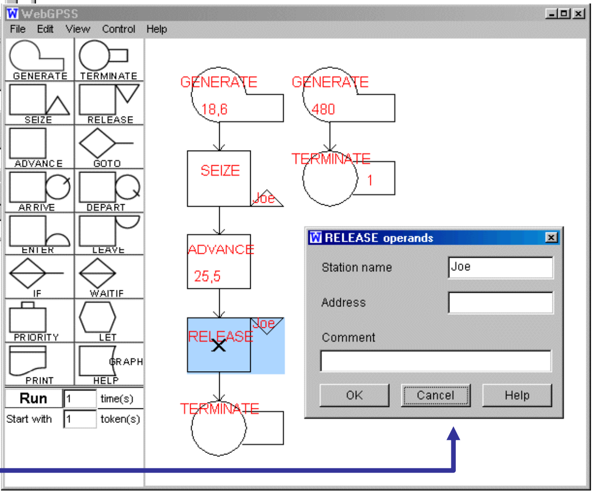
□ Logical

Address:

Comment:

OK Cancel Help

- The station name here is the name of the barber, Joe, by whom the customer wishes to be served.



RELEASE operands

Station name: Joe

Address:

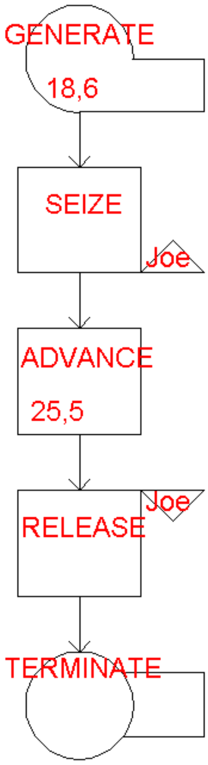
Comment:

OK Cancel Help

- The station name for the RELEASE block is also Joe, the server that the customer will free.

Server names can have from 3 to 6 characters, the first three must be letters, the remaining three letters or digits.

More About SEIZE and RELEASE



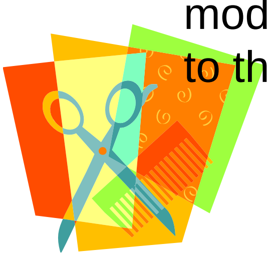
The simplest of GPSS stations, which can serve only one transaction at a time, is known as a **facility** in GPSS.

- When a transaction tries to enter a SEIZE block and the facility is *busy*, then the transaction must wait.
- The transaction waits in the block that it comes to immediately prior to entering the SEIZE block.
- The transaction is then placed on a special waiting list for the facility. The waiting list employs a first-in-first-out (FIFO) queuing discipline.
- A transaction is never refused entry to a RELEASE block.

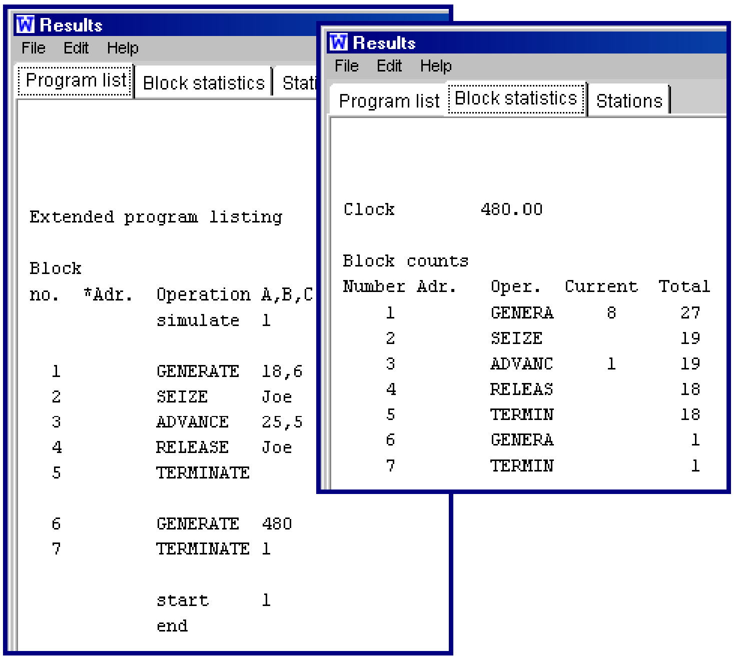
Study Question

Question Suppose that the barber has four chairs for waiting customers. Will this number of chairs be sufficient?

We run our simulation model to find out the answer to this question…



Block Statistics

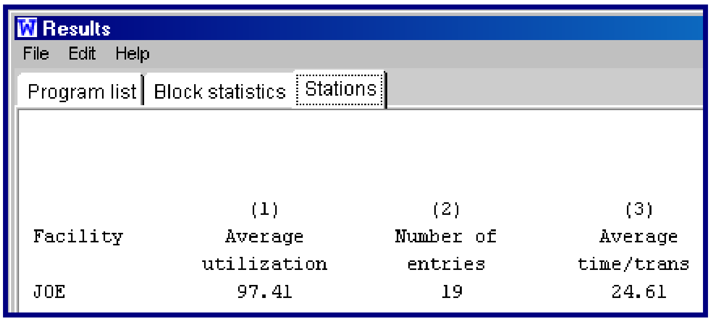


```
Extended program listing

Block
no.   *Adr.   Operation A,B,C
              simulate  1

  1           GENERATE  18,6
  2           SEIZE     Joe
  3           ADVANCE   25,5
  4           RELEASE   Joe
  5           TERMINATE

  6           GENERATE  480
  7           TERMINATE 1

              start     1
              end
```

```
Clock        480.00

Block counts
Number Adr.     Oper.   Current   Total
    1           GENERA      8       27
    2           SEIZE               19
    3           ADVANC      1       19
    4           RELEAS              18
    5           TERMIN              18
    6           GENERA               1
    7           TERMIN               1
```

- **Question #1** How many customers entered the shop during the workday?
- **Question #2** How many were waiting in line at the close of the shop at the end of the day? Are four chairs sufficient?
- **Question #3** How many haircuts were completed by the end of the workday?

Stations Statistics



```
                      (1)            (2)            (3)
Facility            Average        Number of       Average
                  utilization       entries       time/trans
JOE                  97.41            19            24.61
```

- **Question #1** What is the per cent of the total working day that Joe is busy?
- **Question #2** What is the number of customers who have started to get their hair cut by Joe?
- **Question #3** What is the average amount of time that it takes to cut each customer’s hair?
- **Question #4** Average time/trans is busy time divided by the number of entries. How does this fact affect the calculation of the average?

WebGPSS

6

Lesson 6

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

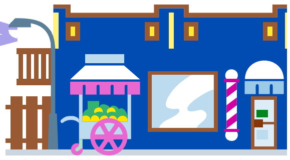
UIN Sunan Kalijaga Yogyakarta

WebGPSS

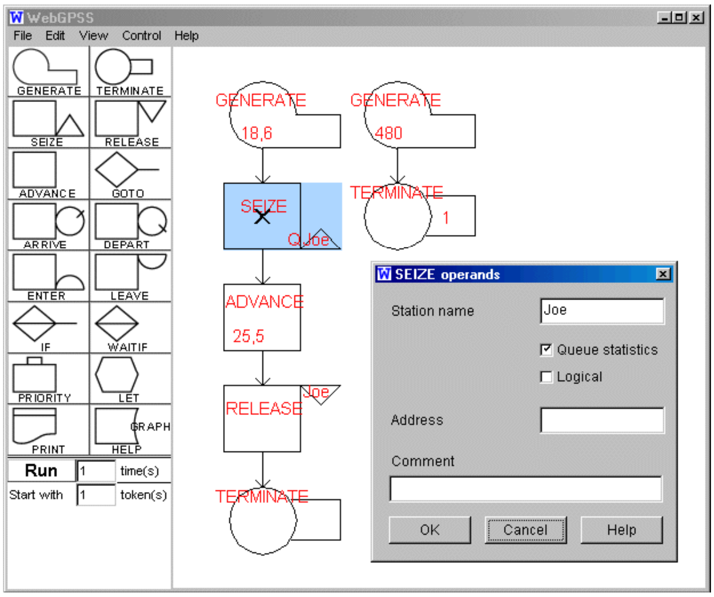
The Lesson 6 Simulation Model

6

- We continue with the model of the previous lesson, but now get more detailed statistics about the waiting line.
- The system is a *small barbershop,* with a single barber, whose name is Joe.
- Customers arrive **randomly**, between 12 and 24 minutes apart, uniformly distributed on this interval.
- The time for a haircut varies between 20 and 30 minutes, uniformly distributed on this interval.
- Joe can serve only one customer at a time, so a waiting line will form if Joe cannot keep up with the customer arrivals.
- The barbershop will close and the simulation will stop after 8 hours of operation.



Obtaining Queue Statistics



SEIZE operands

Station name: Joe

☑ Queue statistics

□ Logical

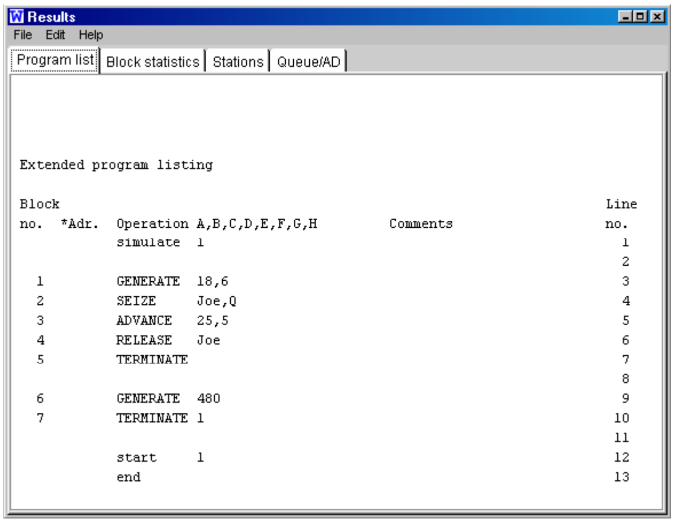
Address:

Comment:

OK Cancel Help

- Double-click on the SEIZE block to open its operand dialog.
- Click on the check-box for *Queue statistics*. This turns on the collection of queue statistics for the station Joe.
- After closing the SEIZE operands dialog by clicking OK, a Q appears in the lower right-hand corner of the SEIZE block rectangle.

Extended Program Listing



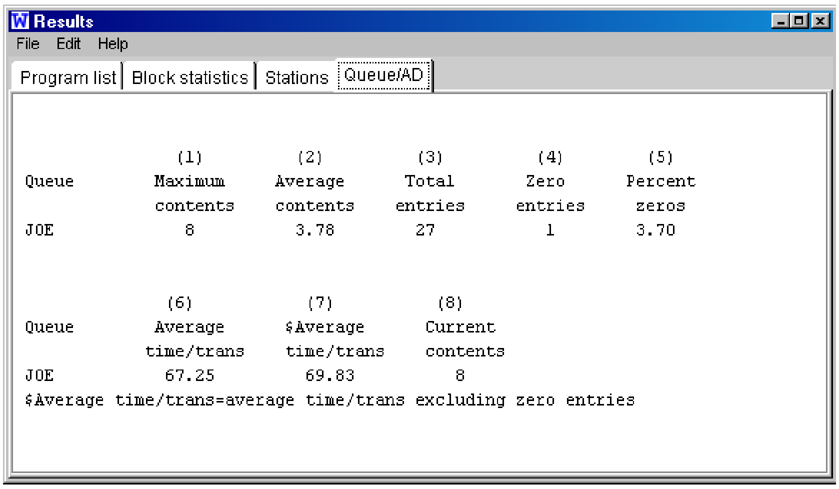
```
Extended program listing

Block                                                        Line
no.  *Adr.  Operation A,B,C,D,E,F,G,H        Comments        no.
            simulate  1                                        1
                                                               2
  1         GENERATE  18,6                                     3
  2         SEIZE     Joe,Q                                    4
  3         ADVANCE   25,5                                     5
  4         RELEASE   Joe                                      6
  5         TERMINATE                                          7
                                                               8
  6         GENERATE  480                                      9
  7         TERMINATE 1                                       10
                                                              11
            start     1                                       12
            end                                               13
```

- Note that the SEIZE block now contains Q as the B operand.
- Block and Station statistics are not changed from the previous model.
- The Q only gathers statistics, which we can study by selecting the new *Queue/AD* tab.

Queue/AD Statistics

6

WebGPSS



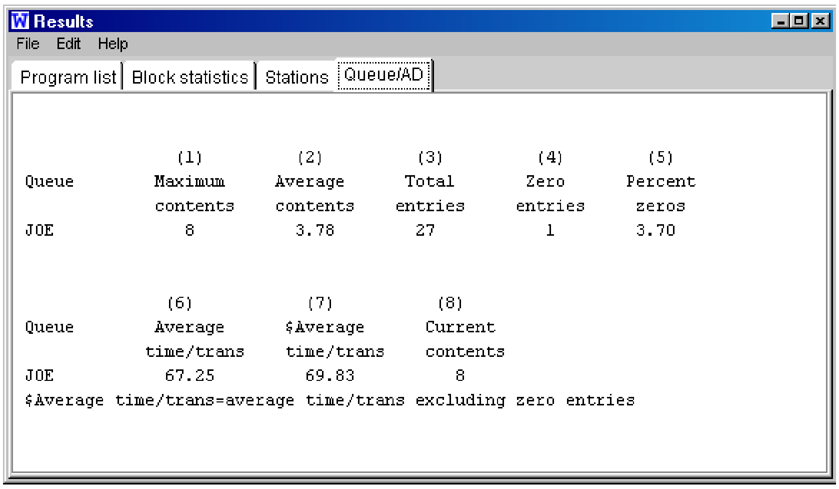
```
               (1)         (2)         (3)         (4)        (5)
Queue        Maximum     Average     Total       Zero       Percent
             contents    contents    entries     entries    zeros
JOE            8           3.78       27           1         3.70


              (6)          (7)         (8)
Queue       Average      $Average    Current
            time/trans   time/trans  contents
JOE           67.25        69.83        8
$Average time/trans=average time/trans excluding zero entries
```

1. **Maximum contents** The highest number of customers that ever waited at any time during the simulation.
2. **Average contents** The total time spend by all customers in the queue divided by the total simulation time.
3. **Total entries** The number of customers who have joined the queue.
4. **Zero entries** The number of customers who have spent zero time in the queue.

Queue/AD Statistics

6

WebGPSS



```
               (1)         (2)         (3)         (4)        (5)
Queue        Maximum     Average     Total       Zero       Percent
             contents    contents    entries     entries    zeros
JOE            8           3.78       27           1         3.70


              (6)          (7)         (8)
Queue       Average      $Average    Current
            time/trans   time/trans  contents
JOE           67.25        69.83        8
$Average time/trans=average time/trans excluding zero entries
```

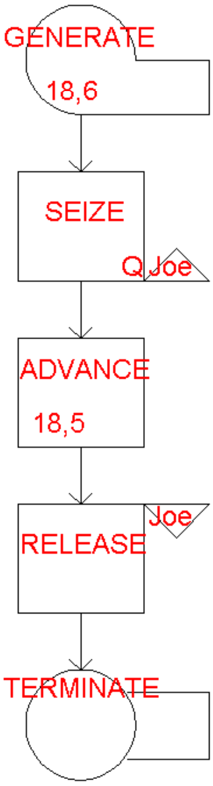
Why are the Queue/AD statistics items number 6 and 7 often **underestimated**?

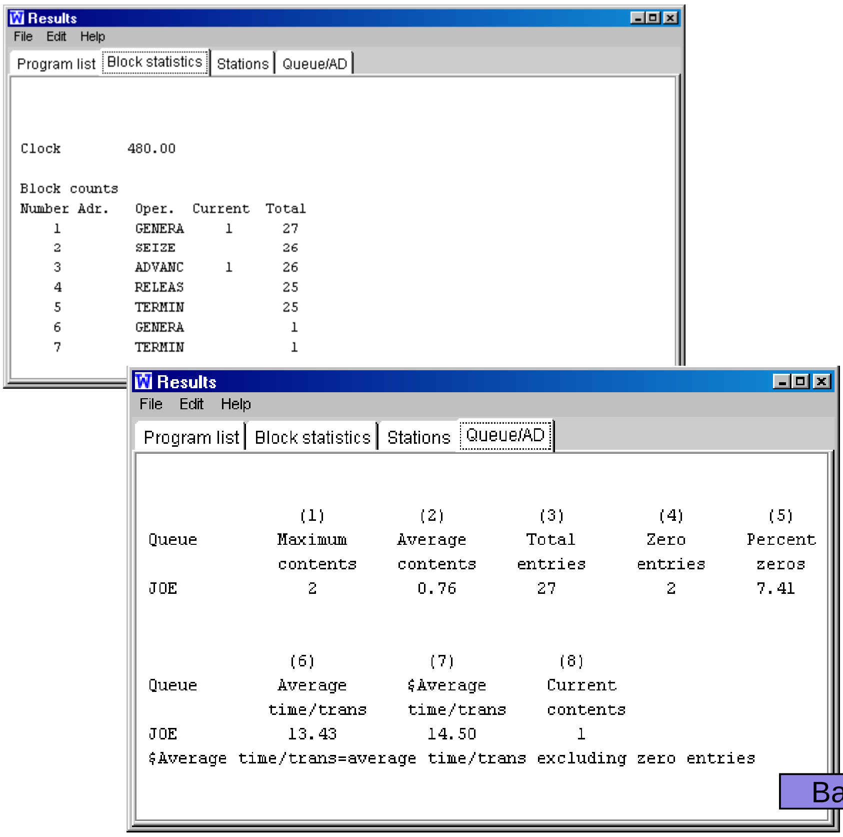
5. **Percent zeros** The percentage ratio of zero to total entries.
6. **Average time/trans** The total time spent by all customers in the queue divided by the total entries into the queue.
7. **$Average time/trans** The average waiting time for those customers that actually had to wait – in other words, *excluding zero entries*.
8. **Current contents** The number of customers who were waiting when the simulation ended.

Discussion Questions

- **Question #1** What is the reason for the large waiting line that developed in this barber shop model?
- **Question #2** Is this the only reason that could cause a waiting line to form?
- **Question #3** What do you think would happen if we changed our barber shop model so that the average time to cut a customer's hair is the same as the average interarrival time to th Results ?
- **Question #4** What do you think would happen if we changed our barber shop model so that it is completely *deterministic*, with both interarrival time and hair-cutting time precisely 18 minutes for every customer? Results

Question 3 Results





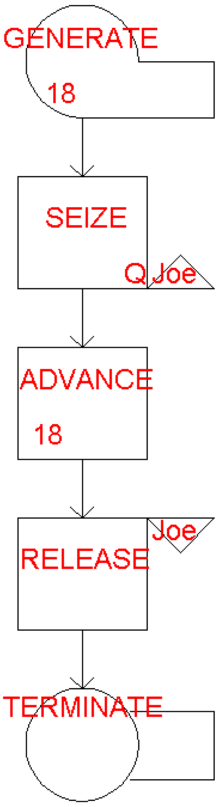
```
Clock        480.00

Block counts
Number Adr.    Oper.   Current   Total
     1         GENERA      1       27
     2         SEIZE               26
     3         ADVANC      1       26
     4         RELEAS              25
     5         TERMIN              25
     6         GENERA               1
     7         TERMIN               1
```

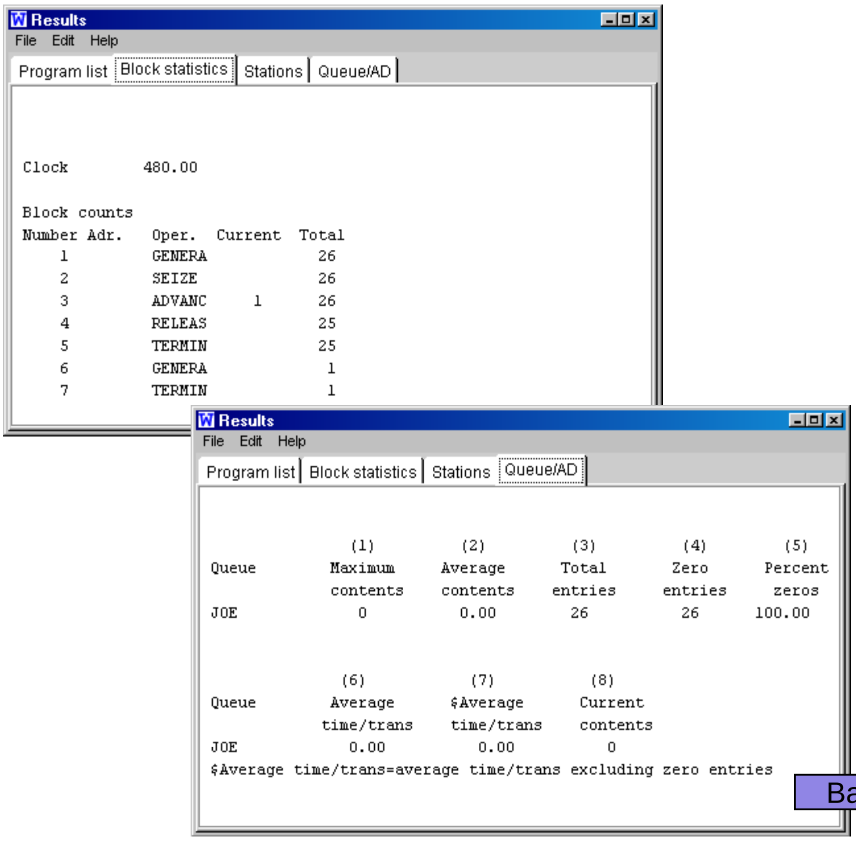
```
Queue          (1)         (2)         (3)         (4)         (5)
             Maximum     Average      Total        Zero      Percent
             contents    contents    entries     entries      zeros
JOE             2          0.76        27           2          7.41


Queue          (6)          (7)          (8)
             Average      $Average     Current
            time/trans   time/trans    contents
JOE           13.43        14.50          1
$Average time/trans=average time/trans excluding zero entries
```

Back

Question 4 Results

WebGPSS





```
Clock        480.00

Block counts
Number Adr.     Oper.  Current  Total
     1          GENERA            26
     2          SEIZE             26
     3          ADVANC     1      26
     4          RELEAS            25
     5          TERMIN            25
     6          GENERA             1
     7          TERMIN             1
```

```
                  (1)        (2)        (3)        (4)        (5)
Queue           Maximum    Average     Total       Zero     Percent
                contents   contents    entries    entries    zeros
JOE                0         0.00        26         26       100.00


                 (6)        (7)          (8)
Queue          Average    $Average     Current
              time/trans  time/trans   contents
JOE              0.00       0.00          0
$Average time/trans=average time/trans excluding zero entries
```

Back

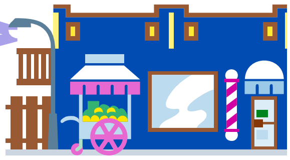
Lesson 7

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

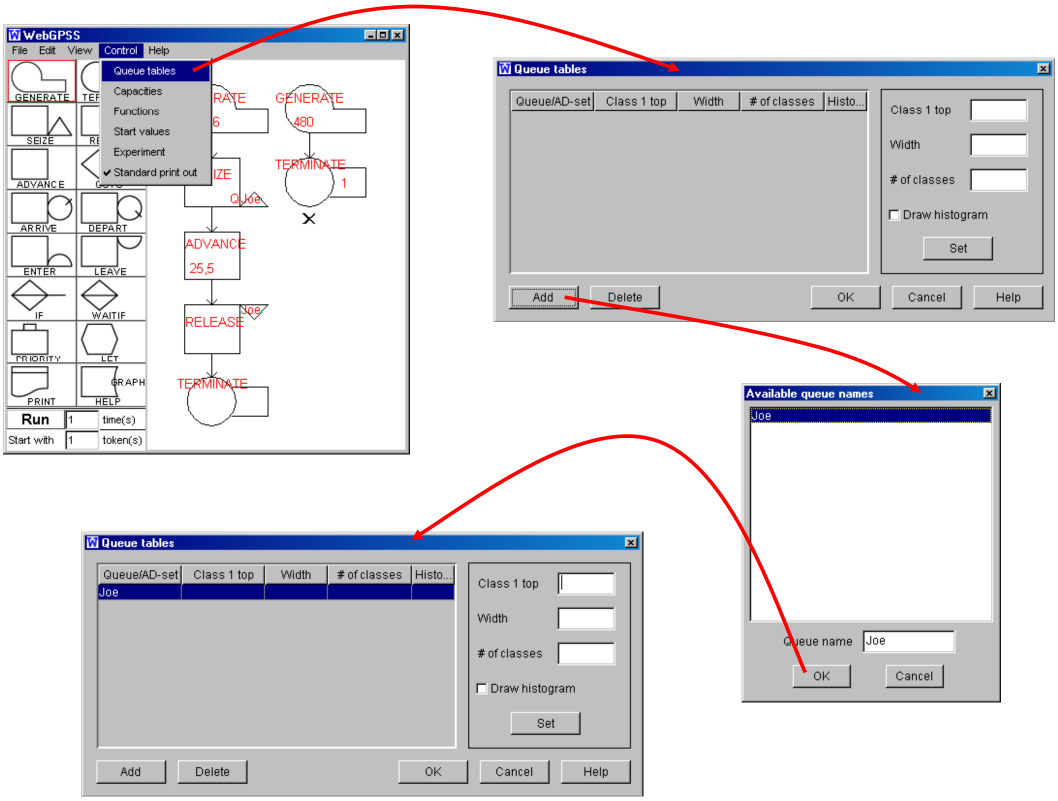
UIN Sunan Kalijaga Yogyakarta

The Lesson 7 Simulation Model



- We continue with the model of the previous two lessons, but now get more detailed waiting line statistics in the form of a queue table.
- The system is a *small barbershop,* with a single barber, whose name is Joe.
- Customers arrive **randomly**, between 12 and 24 minutes apart, uniformly distributed on this interval.
- The time for a haircut varies between 20 and 30 minutes, uniformly distributed on this interval.
- Joe can serve only one customer at a time, so a waiting line will form if Joe cannot keep up with the customer arrivals.
- The barbershop will close and the simulation will stop after 8 hours of operation.

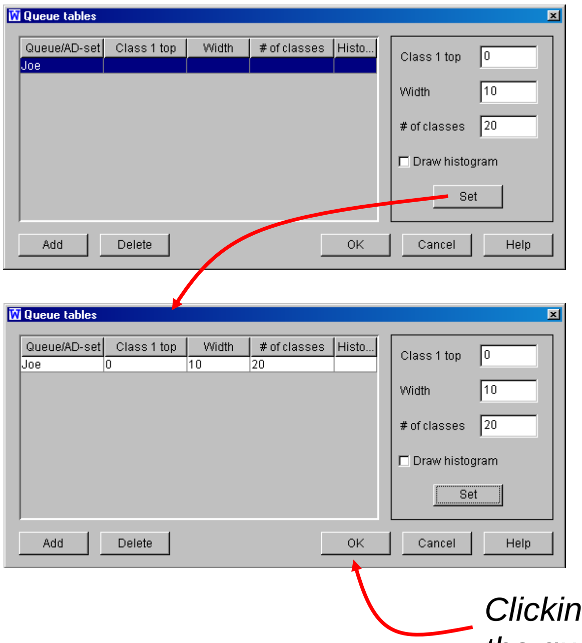
Defining a Queue Table



Queue name: Joe

Queue/AD-set	Class 1 top	Width	# of classes	Histo...
Joe				

Defining a Queue Table



Class 1 top: 0
Width: 10
# of classes: 20

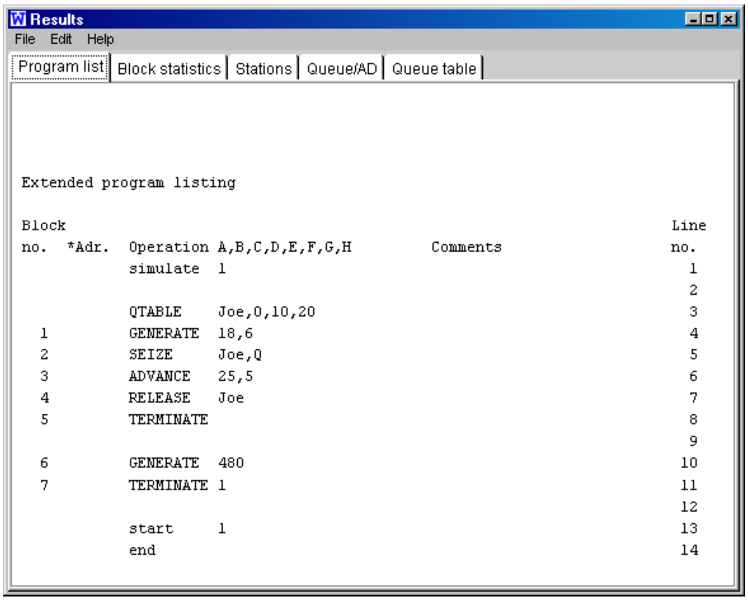
Queue/AD-set	Class 1 top	Width	# of classes	Histo...
Joe	0	10	20	

- **Class 1 Top** The upper limit for the lowest class. All customers waiting 0 minutes, in this case, are shown in this class.
- **Width** The number of time units in each class (except for the lowest and highest classes).
- **# of classes** The maximum number of classes we wish to have in the queue table.

Clicking OK confirms and closes the queue tables dialog.

…0]	**(0…10]**	**(10…20]**	**(20…30]**	**…**	**(170…180]**	**(180…**	**min**
1st	**2nd**	**3rd**	**4th**	**…**	**19th**	**20th**	**class**

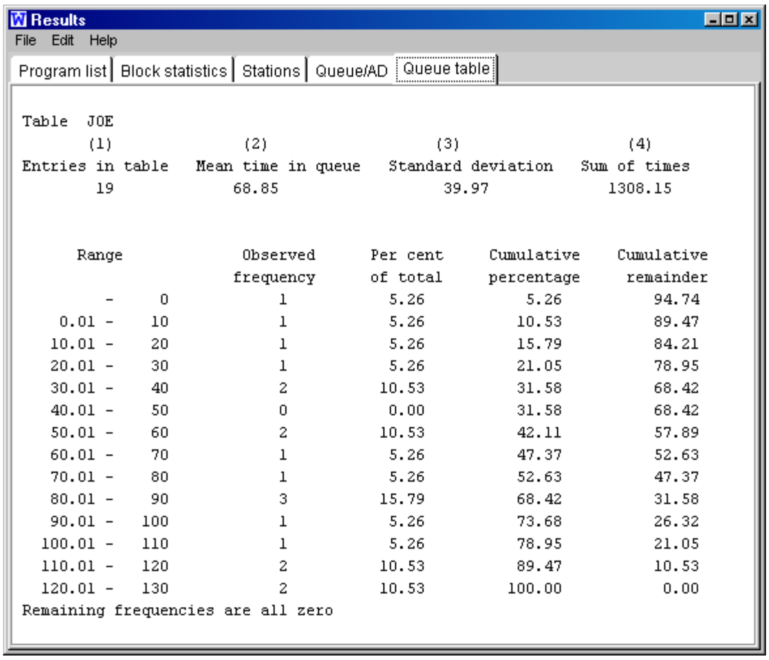
Results – Program List



```
Extended program listing

Block                                                        Line
no.  *Adr.  Operation A,B,C,D,E,F,G,H        Comments        no.
            simulate  1                                         1
                                                                2
            QTABLE    Joe,0,10,20                               3
  1         GENERATE  18,6                                      4
  2         SEIZE     Joe,Q                                     5
  3         ADVANCE   25,5                                      6
  4         RELEASE   Joe                                       7
  5         TERMINATE                                           8
                                                                9
  6         GENERATE  480                                      10
  7         TERMINATE 1                                        11
                                                               12
            start     1                                        13
            end                                                14
```

- We see that the program listing now has a QTABLE control statement appearing prior to the first GENERATE.
- The QTABLE control statement has four operands, A through D.

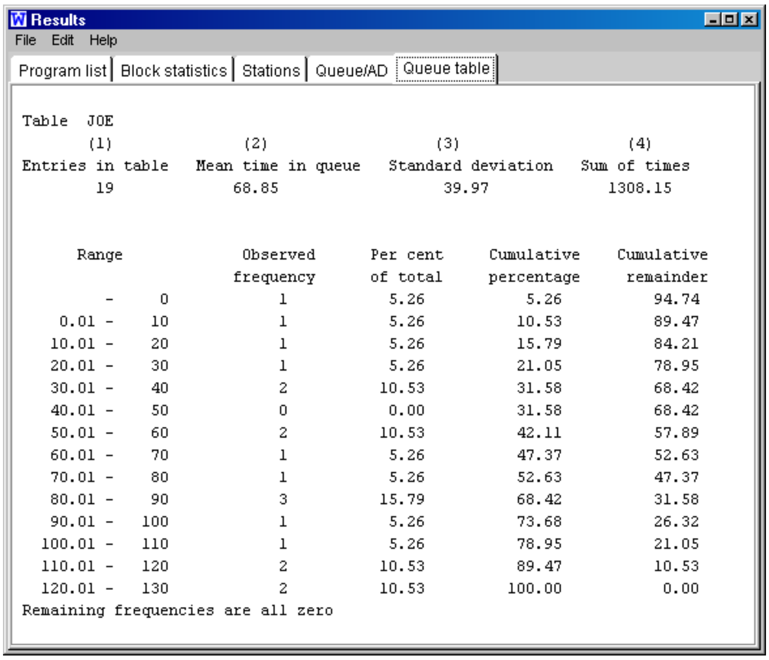
Results – Queue Table



```
Table  JOE
       (1)              (2)                  (3)              (4)
Entries in table  Mean time in queue  Standard deviation  Sum of times
       19               68.85                39.97          1308.15


      Range          Observed    Per cent   Cumulative   Cumulative
                     frequency   of total   percentage   remainder
       -      0          1         5.26        5.26        94.74
    0.01 -   10          1         5.26       10.53        89.47
   10.01 -   20          1         5.26       15.79        84.21
   20.01 -   30          1         5.26       21.05        78.95
   30.01 -   40          2        10.53       31.58        68.42
   40.01 -   50          0         0.00       31.58        68.42
   50.01 -   60          2        10.53       42.11        57.89
   60.01 -   70          1         5.26       47.37        52.63
   70.01 -   80          1         5.26       52.63        47.37
   80.01 -   90          3        15.79       68.42        31.58
   90.01 -  100          1         5.26       73.68        26.32
  100.01 -  110          1         5.26       78.95        21.05
  110.01 -  120          2        10.53       89.47        10.53
  120.01 -  130          2        10.53      100.00         0.00
Remaining frequencies are all zero
```

- **Entries in table** The number of customers that have **FINISHED** waiting.
- **Mean time in queue** The average waiting time for customers that have finished waiting.
- **Standard deviation** A standard measure for describing variations in waiting times in this table.
- **Sum of times** The total time spent by all customers that have finished waiting.

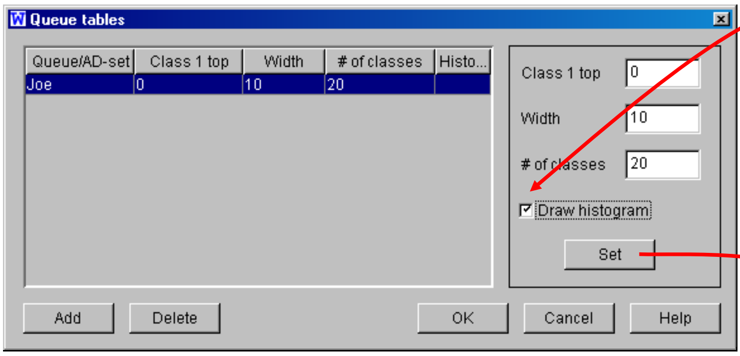
Results – Queue Table



```
Table  JOE
       (1)              (2)                 (3)              (4)
Entries in table   Mean time in queue   Standard deviation   Sum of times
        19               68.85                39.97            1308.15


      Range               Observed     Per cent     Cumulative    Cumulative
                          frequency    of total     percentage     remainder
          -      0            1           5.26          5.26         94.74
     0.01 -     10            1           5.26         10.53         89.47
    10.01 -     20            1           5.26         15.79         84.21
    20.01 -     30            1           5.26         21.05         78.95
    30.01 -     40            2          10.53         31.58         68.42
    40.01 -     50            0           0.00         31.58         68.42
    50.01 -     60            2          10.53         42.11         57.89
    60.01 -     70            1           5.26         47.37         52.63
    70.01 -     80            1           5.26         52.63         47.37
    80.01 -     90            3          15.79         68.42         31.58
    90.01 -    100            1           5.26         73.68         26.32
   100.01 -    110            1           5.26         78.95         21.05
   110.01 -    120            2          10.53         89.47         10.53
   120.01 -    130            2          10.53        100.00          0.00
Remaining frequencies are all zero
```

- **Range** Classes expressed in time intervals.
- **Observed frequency** The number of customers in each class.
- **Per cent of total** Observed frequency as a per cent of the total number of customers who have finished waiting.
- **Cumulative percentage** Added cumulatively to help answering questions like "What percent of the customers waited at most *x* minutes?"
- **Cumulative remainder** Helps in answering questions like "What percent of the customers waited more than *x* minutes?

Obtaining a Queue Table Histogram

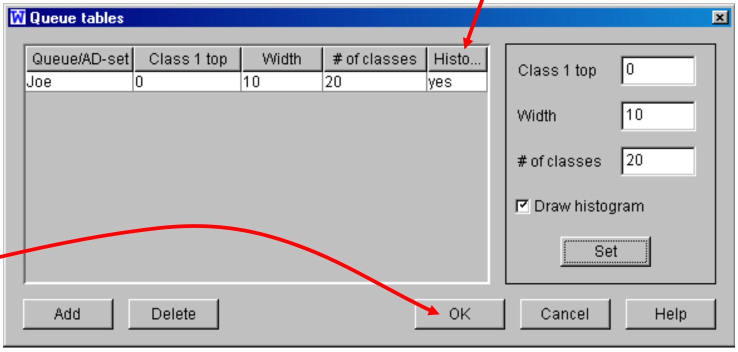


Queue/AD-set	Class 1 top	Width	# of classes	Histo...
Joe	0	10	20	

Class 1 top: 0
Width: 10
# of classes: 20
☑ Draw histogram
Set

First, click on the *Draw histogram* checkbox.

Then, click on the Set button.

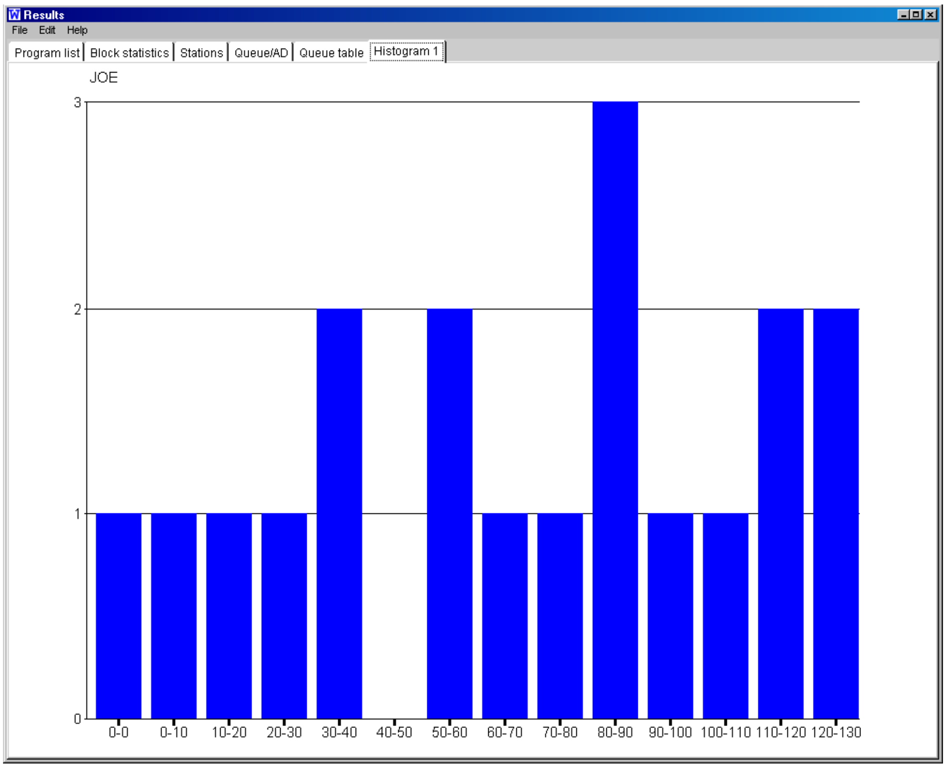


Queue/AD-set	Class 1 top	Width	# of classes	Histo...
Joe	0	10	20	yes

Class 1 top: 0
Width: 10
# of classes: 20
☑ Draw histogram
Set

Finally, click on OK to confirm and close the Queue tables dialog.

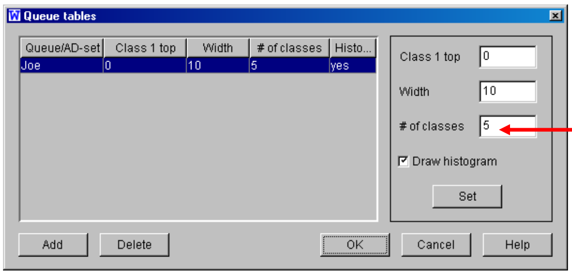
Results – Histogram



This window can be resized by dragging its corners!

Queue Table Overflow

- What happens if we do not specify enough classes in the queue table definition?



Queue/AD-set	Class 1 top	Width	# of classes	Histo...
Joe	0	10	5	yes

Class 1 top 0

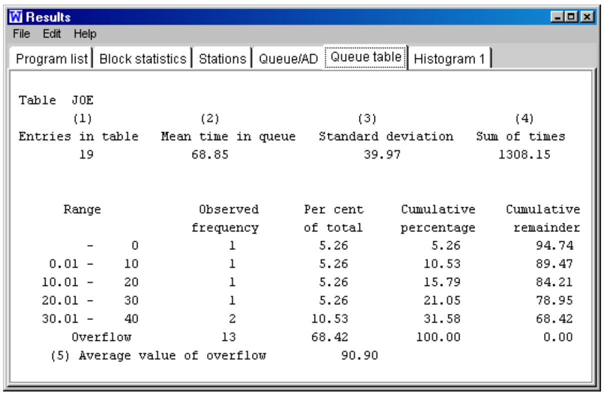
Width 10

# of classes 5

☑ Draw histogram

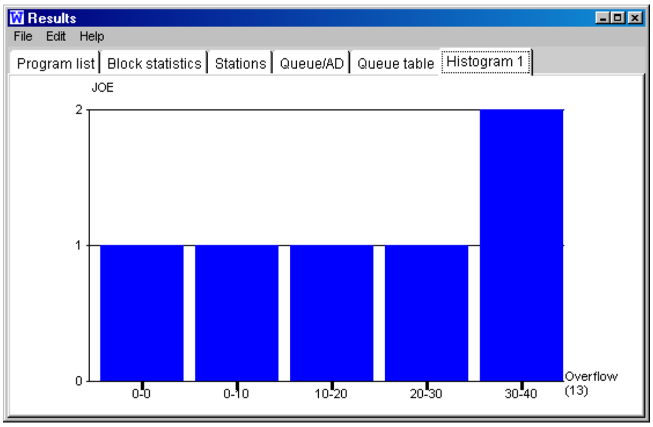
Suppose that we indicate the number of classes as 5, instead of the original 20.

Queue Table Overflow



```
Table  JOE
       (1)              (2)                 (3)                 (4)
Entries in table   Mean time in queue   Standard deviation   Sum of times
        19              68.85               39.97              1308.15


     Range              Observed      Per cent    Cumulative    Cumulative
                        frequency     of total    percentage    remainder
        -      0            1           5.26         5.26         94.74
    0.01 -    10            1           5.26        10.53         89.47
   10.01 -    20            1           5.26        15.79         84.21
   20.01 -    30            1           5.26        21.05         78.95
   30.01 -    40            2          10.53        31.58         68.42
       Overflow            13          68.42       100.00          0.00
   (5) Average value of overflow            90.90
```

We see that 13 customers waited longer that 40 minutes. These 13 customers waited 90.9 minutes on average.



The histogram also indicates that there was an *overflow* of 13 customers.

WebGPSS

8

Lesson 8

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

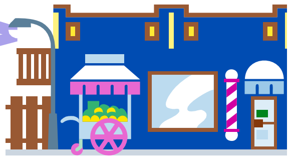
UIN Sunan Kalijaga Yogyakarta

WebGPSS

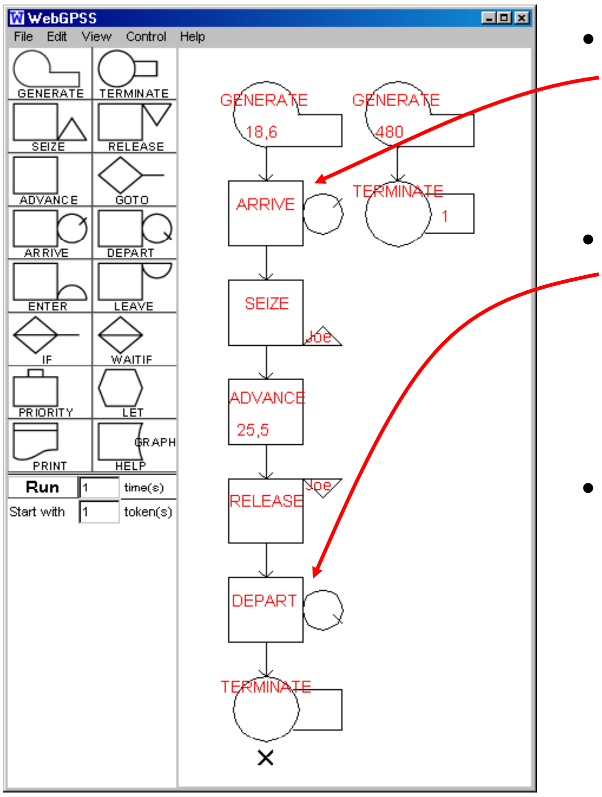
8

The Lesson 8 Simulation Model

- We continue with the barbershop model, but now measure the *total time* that customers spend in the shop.
- The system is a *small barbershop,* with a single barber, whose name is Joe.
- Customers arrive **randomly**, between 12 and 24 minutes apart, uniformly distributed on this interval.
- The time for a haircut varies between 20 and 30 minutes, uniformly distributed on this interval.
- Joe can serve only one customer at a time, so a waiting line will form if Joe cannot keep up with the customer arrivals.
- The barbershop will close and the simulation will stop after 8 hours of operation.

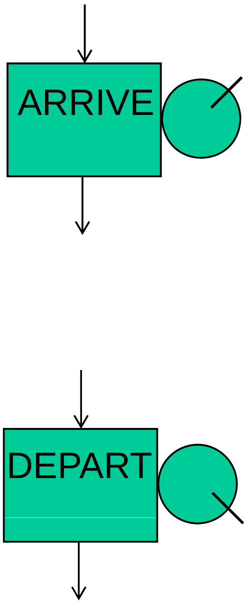


The ARRIVE and DEPART Blocks



- We insert an ARRIVE block between the GENERATE and SEIZE block to mark the arrival of a customer.
- We insert a DEPART block between the RELEASE and TERMINATE block to mark the departure of a customer from the barbershop.
- We observe that the ARRIVE and DEPART blocks are *mirror images* of one another.

The ARRIVE and DEPART Blocks

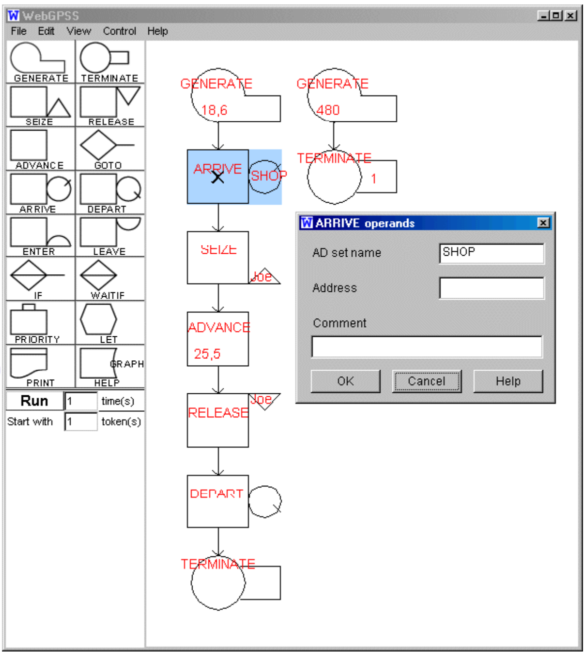


One can think of the ARRIVE block as equipping *each* customer with a stopwatch that is started when the customer enters the ARRIVE block.

The stopwatch is stopped when the customer enters the DEPART block.

The small circles on these blocks resemble a stopwatch with the “slash” being the second hand that has moved with the passage of time.

ARRIVE and DEPART Operands

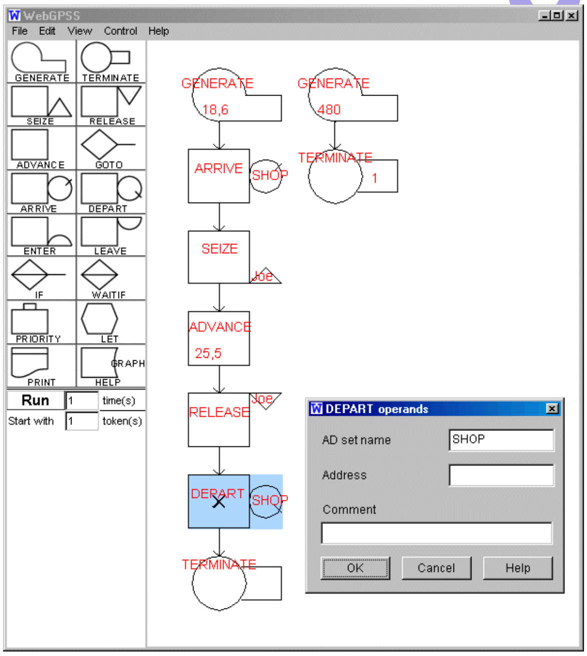


ARRIVE operands

AD set name: SHOP

Address:

Comment:



DEPART operands

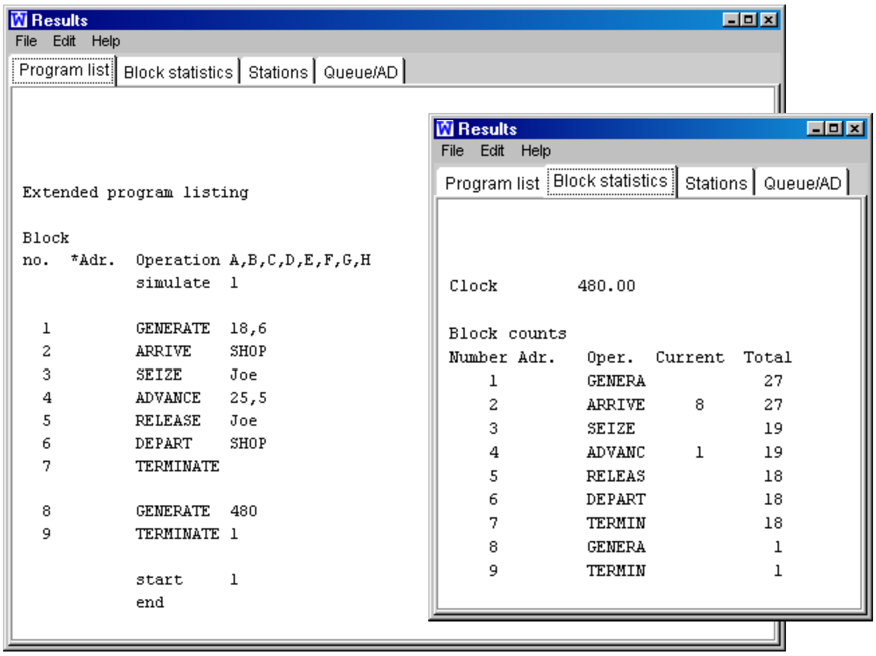
AD set name: SHOP

Address:

Comment:

- We double-click the ARRIVE block and specify the AD set name as SHOP, denoting an arrival at the barbershop.
- We double-click the DEPART block and again specify the AD set name as SHOP, denoting a departure from the barbershop.

Results – Block Statistics

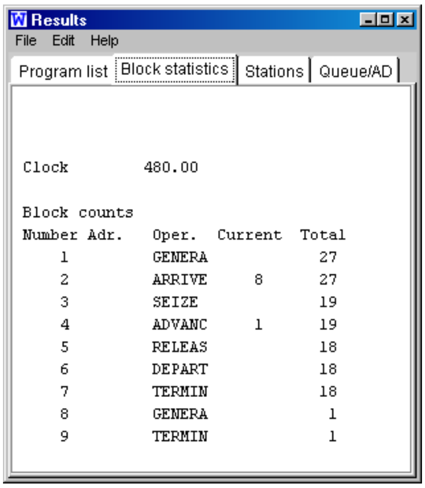


```
Extended program listing

Block
no.  *Adr.   Operation A,B,C,D,E,F,G,H
             simulate  1

  1          GENERATE  18,6
  2          ARRIVE    SHOP
  3          SEIZE     Joe
  4          ADVANCE   25,5
  5          RELEASE   Joe
  6          DEPART    SHOP
  7          TERMINATE

  8          GENERATE  480
  9          TERMINATE 1

             start     1
             end
```

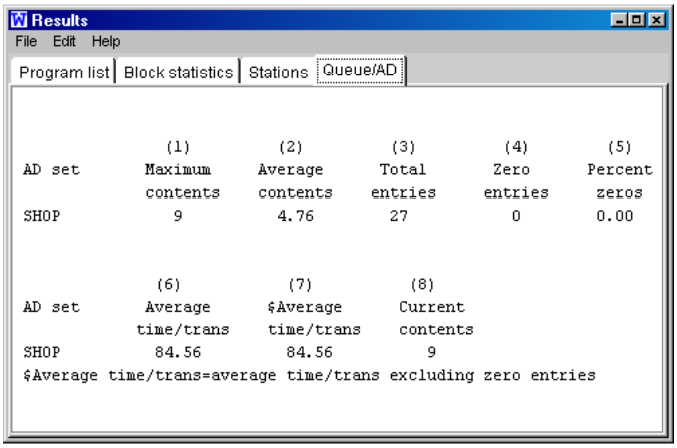
```
Clock        480.00

Block counts
Number Adr.    Oper.   Current  Total
    1          GENERA              27
    2          ARRIVE     8        27
    3          SEIZE               19
    4          ADVANC     1        19
    5          RELEAS              18
    6          DEPART              18
    7          TERMIN              18
    8          GENERA               1
    9          TERMIN               1
```

- Note in the program listing that the AD set name is the A operand for both the ARRIVE and DEPART blocks.
- **Question** Why do we find that the 8 customers waiting at closing time now wait in the ARRIVE block?

Results – Queue/AD Statistics



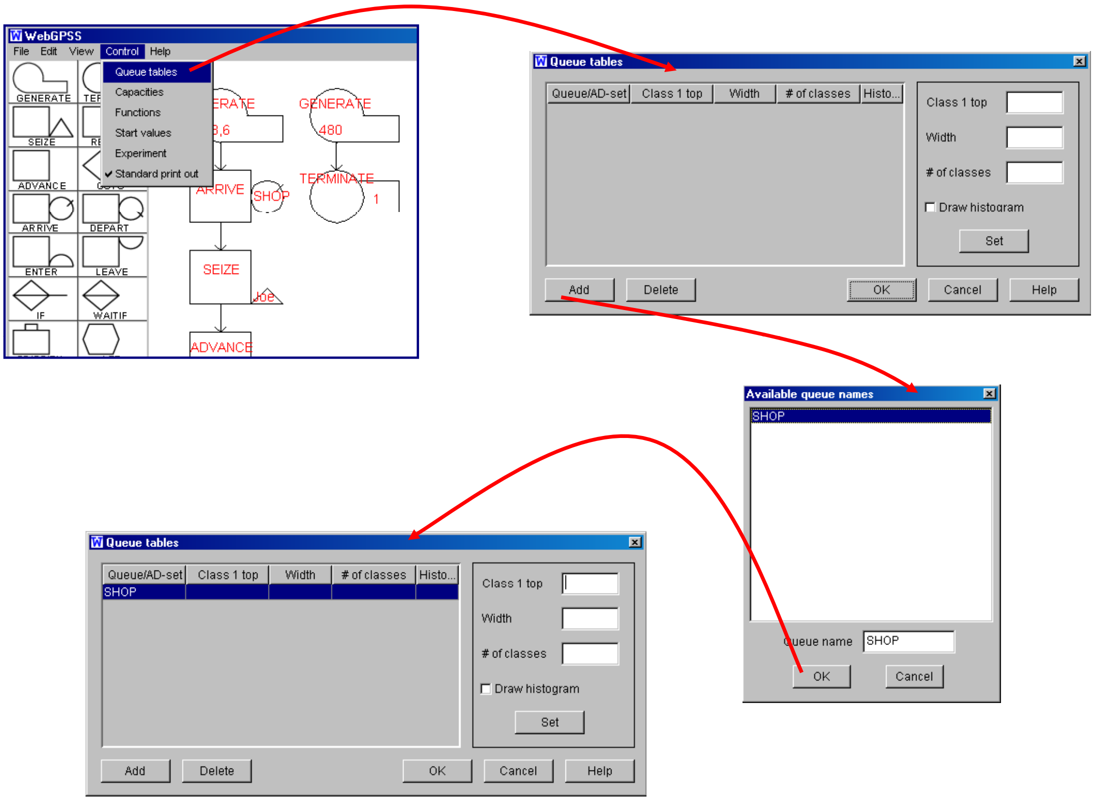
```
Clock         480.00

Block counts
Number Adr.    Oper.   Current   Total
    1          GENERA              27
    2          ARRIVE     8        27
    3          SEIZE               19
    4          ADVANC     1        19
    5          RELEAS              18
    6          DEPART              18
    7          TERMIN              18
    8          GENERA               1
    9          TERMIN               1
```



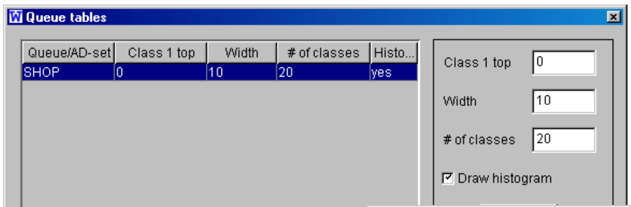
```
             (1)         (2)        (3)        (4)        (5)
AD set      Maximum     Average     Total      Zero      Percent
            contents    contents    entries    entries    zeros
SHOP          9          4.76        27         0         0.00


             (6)          (7)          (8)
AD set      Average     $Average     Current
           time/trans   time/trans   contents
SHOP         84.56        84.56         9
$Average time/trans=average time/trans excluding zero entries
```

- **Question #1** What is the *average time/transaction*, 84.56 minutes, a measure of in this model?
- **Question #2** What is the meaning of *current contents* in the context of this barbershop model?
- **Question #3** Why are there no zero entries?
- **Question #4** In the previous lesson, we saw that the average waiting time for a customer is 67 minutes. Why is the average time spent in the shop (85 minutes) only about 18 minutes longer that the average waiting time?

Defining a QTABLE for an AD Set



Defining a QTABLE for an A/D Set



Queue/AD-set	Class 1 top	Width	# of classes	Histo...
SHOP	0	10	20	yes

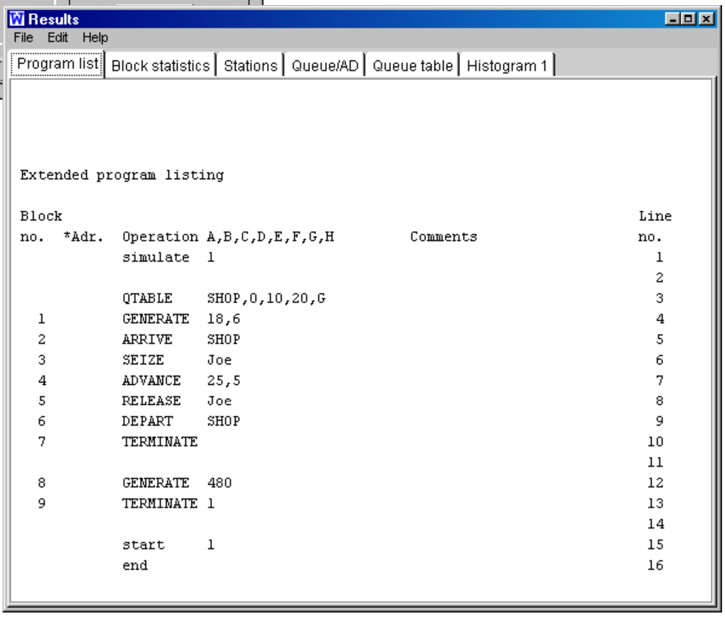
Class 1 top: 0

Width: 10

# of classes: 20

☑ Draw histogram

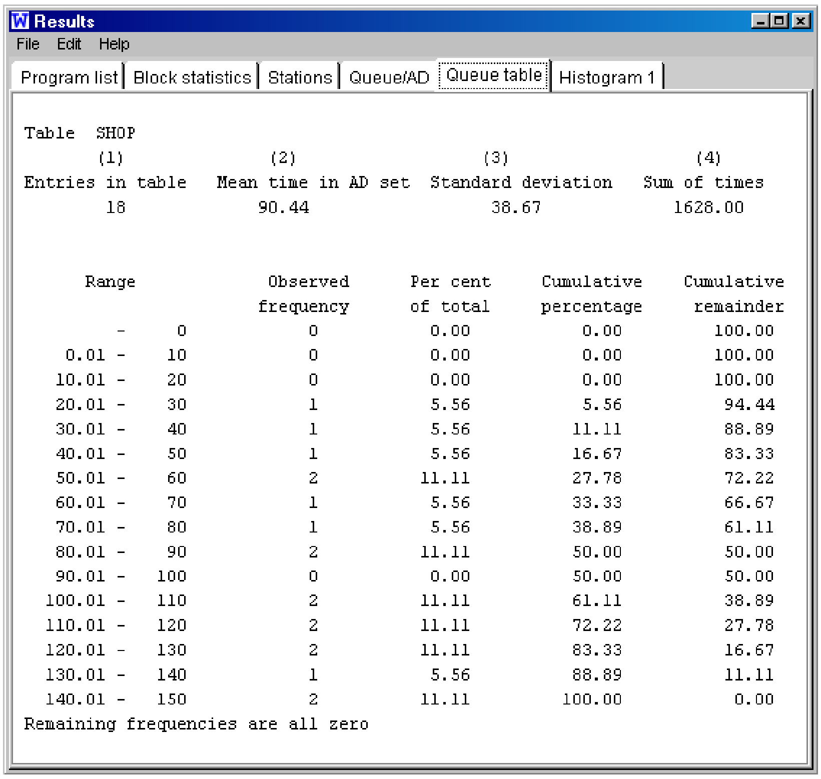
- We define the QTABLE with the same values as in the previous lesson.



```
Extended program listing

Block                                                                      Line
no.   *Adr.  Operation A,B,C,D,E,F,G,H        Comments                       no.
             simulate  1                                                      1
                                                                              2
             QTABLE    SHOP,0,10,20,G                                         3
  1          GENERATE  18,6                                                   4
  2          ARRIVE    SHOP                                                   5
  3          SEIZE     Joe                                                    6
  4          ADVANCE   25,5                                                   7
  5          RELEASE   Joe                                                    8
  6          DEPART    SHOP                                                   9
  7          TERMINATE                                                       10
                                                                             11
  8          GENERATE  480                                                   12
  9          TERMINATE 1                                                     13
                                                                             14
             start     1                                                     15
             end                                                             16
```

- After running the model, we see that a QTABLE control statement for the AD set SHOP appears in the program listing.

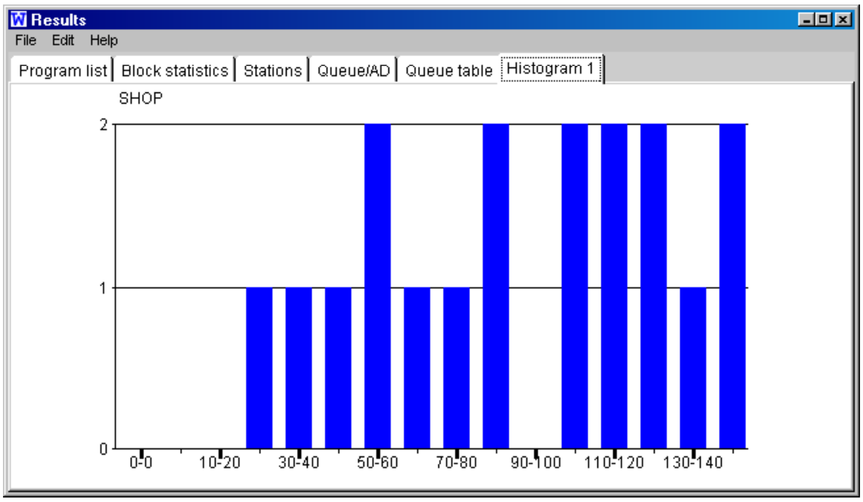
QTABLE for SHOP



```
Table  SHOP
         (1)               (2)              (3)              (4)
Entries in table   Mean time in AD set  Standard deviation  Sum of times
         18              90.44             38.67            1628.00


     Range            Observed     Per cent    Cumulative    Cumulative
                      frequency    of total    percentage    remainder
          -     0         0          0.00         0.00         100.00
     0.01 -    10         0          0.00         0.00         100.00
    10.01 -    20         0          0.00         0.00         100.00
    20.01 -    30         1          5.56         5.56          94.44
    30.01 -    40         1          5.56        11.11          88.89
    40.01 -    50         1          5.56        16.67          83.33
    50.01 -    60         2         11.11        27.78          72.22
    60.01 -    70         1          5.56        33.33          66.67
    70.01 -    80         1          5.56        38.89          61.11
    80.01 -    90         2         11.11        50.00          50.00
    90.01 -   100         0          0.00        50.00          50.00
   100.01 -   110         2         11.11        61.11          38.89
   110.01 -   120         2         11.11        72.22          27.78
   120.01 -   130         2         11.11        83.33          16.67
   130.01 -   140         1          5.56        88.89          11.11
   140.01 -   150         2         11.11       100.00           0.00
Remaining frequencies are all zero
```

- **Question #1** Why is the observed frequency zero for the first three rows of this table?
- **Question #2** What percentage of the customers had to wait at most 30 minutes?
- **Question #3** What percentage of the customers had to wait more than one hour?

Results – Histogram



Question How many customers had to wait more than 2 hours and 20 minutes?

Lesson 9

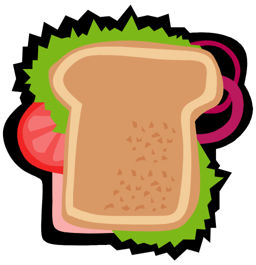
Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

The Lesson 9 Simulation Model

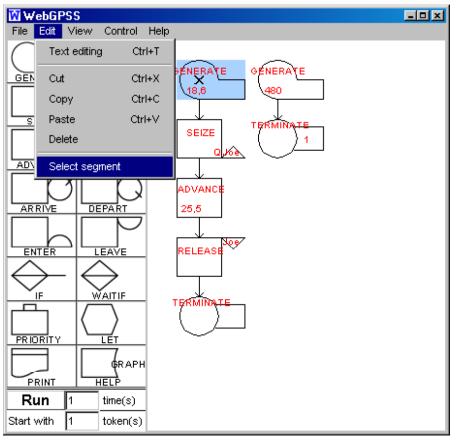
- We now turn our attention to a “7-Eleven” store.
- There is one salesperson, called Joe, but there are *two different types of customer*.
- Newspaper buyers arrive more frequently than food buyers but need a shorter service time.



	IAT	Service Time
Newspaper buyers	90 to 150 seconds	15 to 75 seconds
Food buyers	4 to 8 minutes	90 to 390 seconds

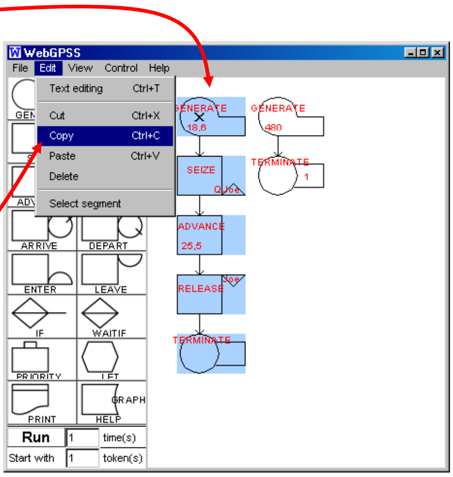
- Joe can service only one customer at a time.
- We will run our simulation for a time of 2 hours.

Copying a Segment

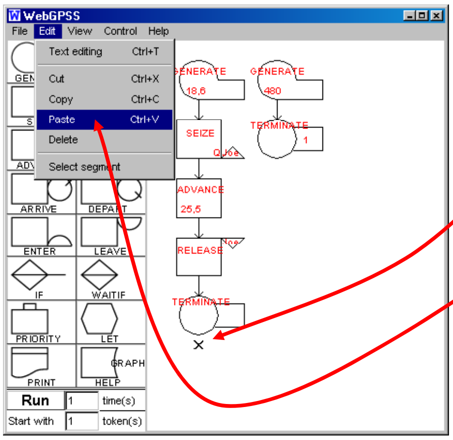


- This program can be built from a previous barbershop model, PRO08, which we open.
- Our 7-Eleven needs two customer segments, so we need to make a copy of the existing customer segment.
- We click on the GENERATE block so it is marked in blue.
- We then click on *Edit/Select segment* from the top menu. The entire segment is then marked in blue.

- We next click on Edit/Copy.



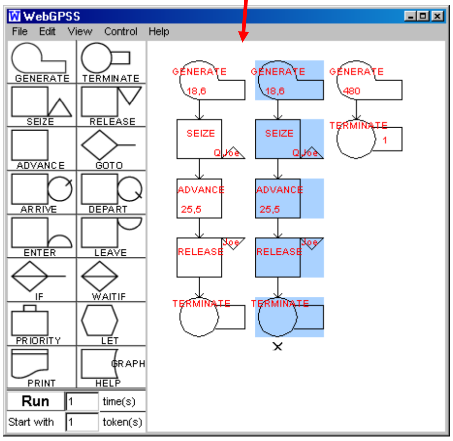
Copying a Segment



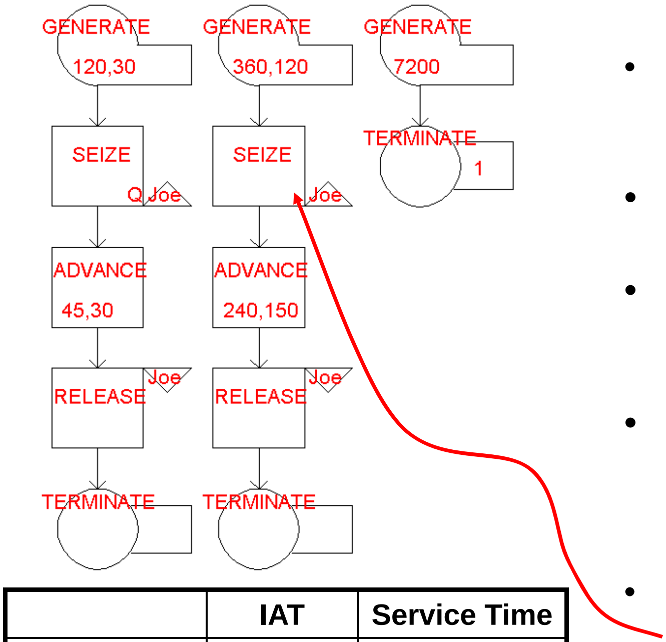
- We next place an X below the TERMINATE block by clicking just below this block.

- Then we click on Edit/Paste from the top menu.

- The result is two identical customer segments.



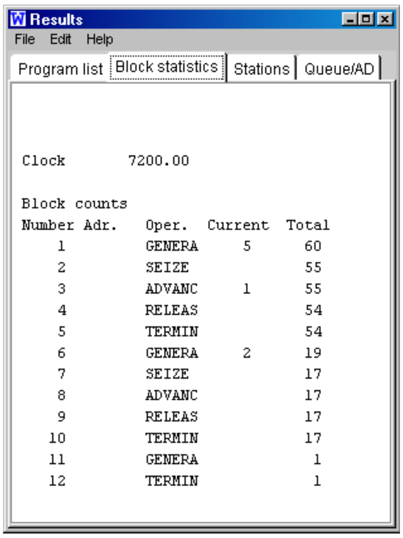
Specifying the Operands



	IAT	Service Time
Newspaper buyers	90 to 150 seconds	15 to 75 seconds
Food buyers	4 to 8 minutes	90 to 390 seconds

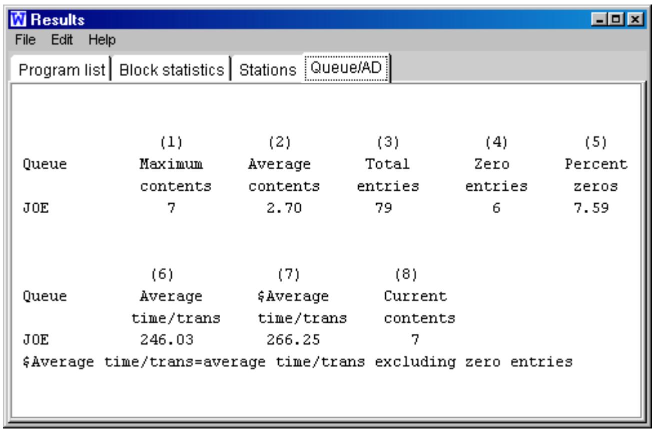
- We enter the operands directly, since we know the syntax for these blocks.
- **Question #1** What is the unit of time for this simulation?
- **Question #2** Which segment is used for the newspaper buyers?
- **Question #3** Why does the GENERATE block for the stop segment have an operand of 7200?
- We can delete the Q in the second SEIZE block, as we have already requested queue statistics in the first SEIZE block.

Results – Block and Queue Statistics



```
Clock        7200.00

Block counts
Number Adr.    Oper.   Current  Total
     1         GENERA      5      60
     2         SEIZE              55
     3         ADVANC      1      55
     4         RELEAS             54
     5         TERMIN             54
     6         GENERA      2      19
     7         SEIZE              17
     8         ADVANC             17
     9         RELEAS             17
    10         TERMIN             17
    11         GENERA              1
    12         TERMIN              1
```

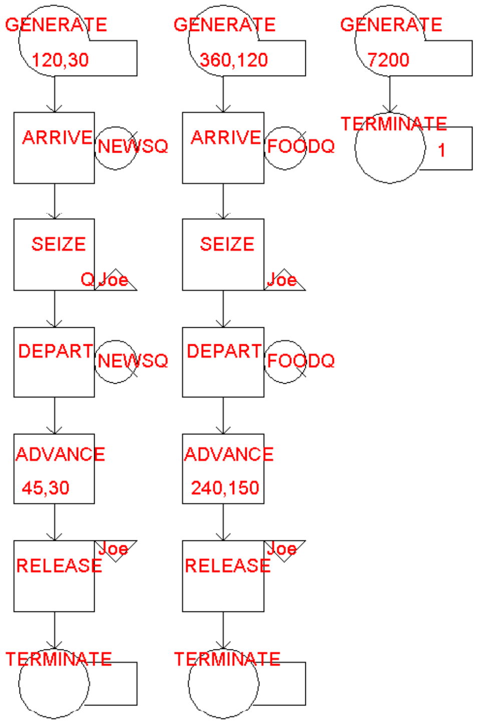
- **Question** At the end of the simulation, how many newspaper buyers and how many food buyers are waiting?
- Note that although the newspaper and food buyers wait in *different blocks*, they join a *single waiting line* and are served FIFO by Joe.



```
               (1)         (2)         (3)         (4)        (5)
Queue        Maximum     Average      Total       Zero      Percent
             contents    contents    entries     entries     zeros
JOE             7          2.70         79          6         7.59


               (6)          (7)          (8)
Queue        Average     $Average     Current
             time/trans  time/trans   contents
JOE           246.03      266.25          7
$Average time/trans=average time/trans excluding zero entries
```

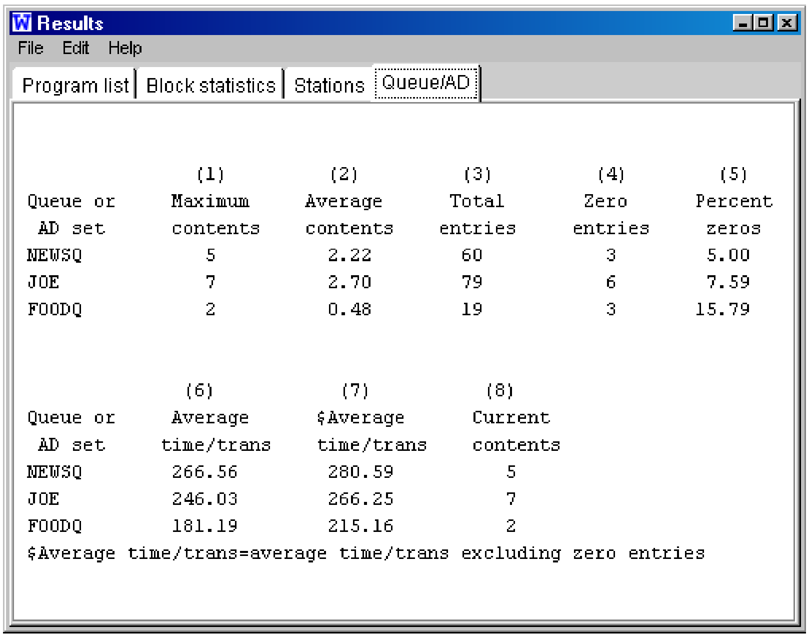
- The queue statistics refer to *all* customers, and make no distinction between newspaper buyers and food buyers.

Separate Queue Statistics



- It may be of interest to determine separate queue statistics for newspaper buyers and food buyers.
- Arrive-Depart block pairs for each of the customer segments allow the capture of separate queue statistics.
- **Question #1** What does the ARRIVE block accomplish in each segment?
- **Question #2** What does the DEPART block accomplish in each segment?

Results – AD Statistics



```
                (1)        (2)         (3)        (4)        (5)
Queue or      Maximum    Average      Total      Zero      Percent
 AD set       contents   contents    entries    entries     zeros
NEWSQ           5          2.22        60          3         5.00
JOE             7          2.70        79          6         7.59
FOODQ           2          0.48        19          3        15.79


               (6)         (7)          (8)
Queue or     Average     $Average     Current
 AD set     time/trans   time/trans   contents
NEWSQ         266.56      280.59        5
JOE           246.03      266.25        7
FOODQ         181.19      215.16        2
$Average time/trans=average time/trans excluding zero entries
```

Question #1 Why might the newspaper buyers be somewhat upset when visiting Joe’s 7-Eleven store?

Question #2 What are some ways that Joe could possibly improve the situation for his newspaper customers?

Lesson 10

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

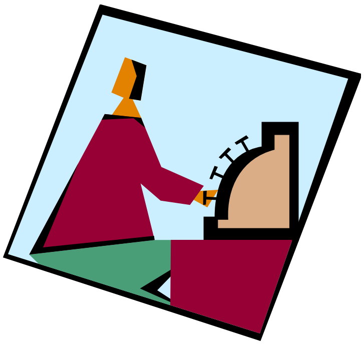
UIN Sunan Kalijaga Yogyakarta

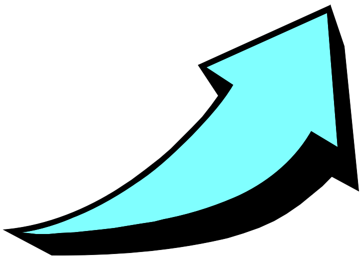
The Lesson 10 Simulation Models

- We have up to this point studied only systems with a *single* station.
- This lesson will look at three different systems with *stations in series*. A transaction will first be serviced by one station and then proceed to another station for additional service.
- The models to be studied include:
  - Joe's *barbershop*, in which customers pay a cashier after getting their haircut.
  - An *engineering shop*, in which parts first use a lathe and then use a cutter during the production process.
  - A *small store* with two workers.

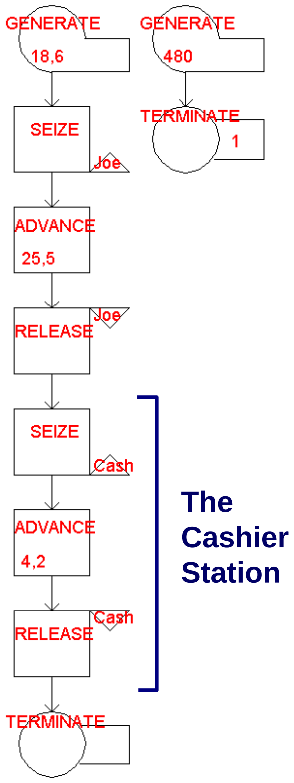
The Barbershop with a Cashier

- In some countries it is common to have a cashier to whom customers make payment following their services from the barber.
- In GPSS this can be modeled by having two stations that work in series, i.e. one after the other.

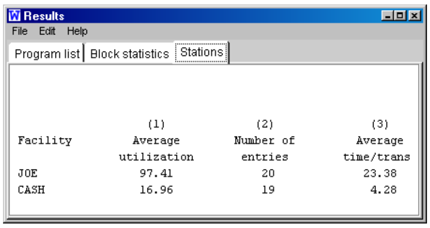




The Barbershop with a Cashier



- Time units are assumed to be minutes.
- We insert SEIZE, ADVANCE, and RELEASE for the cashier station, called **Cash**.
- Note that each new station in a model must have a unique name, different from every other station.
- **Question** How much time is required to make payment?
- **Question** If a waiting line were to form in front of the cashier, in what block would the customers wait?
- In the stations statistics, we now have two lines, one line for each of the two stations.



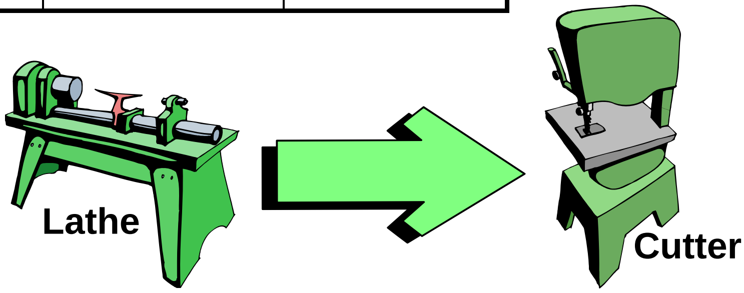
```
                 (1)            (2)          (3)
Facility       Average       Number of     Average
             utilization      entries     time/trans
JOE             97.41           20          23.38
CASH            16.96           19           4.28
```

An Engineering Shop

- We have two machines, a lathe and a cutter, that produce two parts, A and B.
- Prices are such that all production can be sold.
- There are 500 minutes available each workday.
- Machine set-up times are negligible.

	Lathe Time (minutes)	Cutter Time (minutes)	Gross Profit per Unit
Part A	20 ± 5	10 ± 5	$2
Part B	10 ± 3	30 ± 10	$3

PROBLEM Determine a daily production schedule that maximizes gross profit and test its feasibility.



Linear Programming Solution

…uses average production times, ignoring random variations

Let Q_A = quantity of A to produce each day
Q_B = quantity of B to produce each day

Maximize Daily Gross Profit: $2Q_A + 3Q_B$

Subject to the constraints:

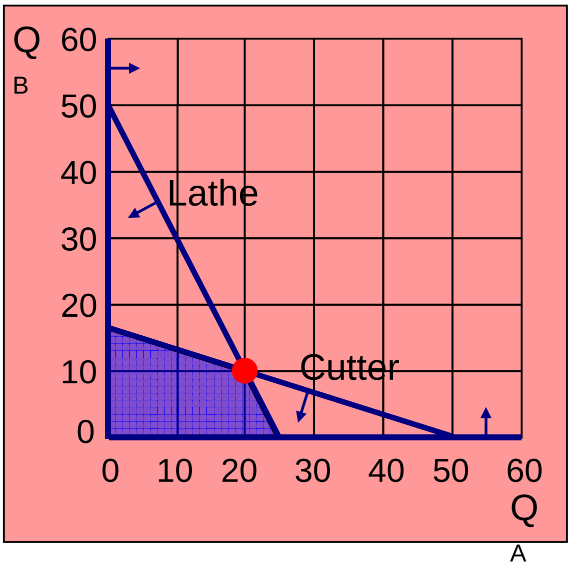
$Q_A >= 0$; $Q_B >= 0$ (non-negativity)

$20Q_A + 10Q_B <= 500$ (Lathe)

$10Q_A + 30Q_B <= 500$ (Cutter)

Optimal Solution: ●

The daily gross profit will be optimized at $70 by producing 20 A and 10 B.

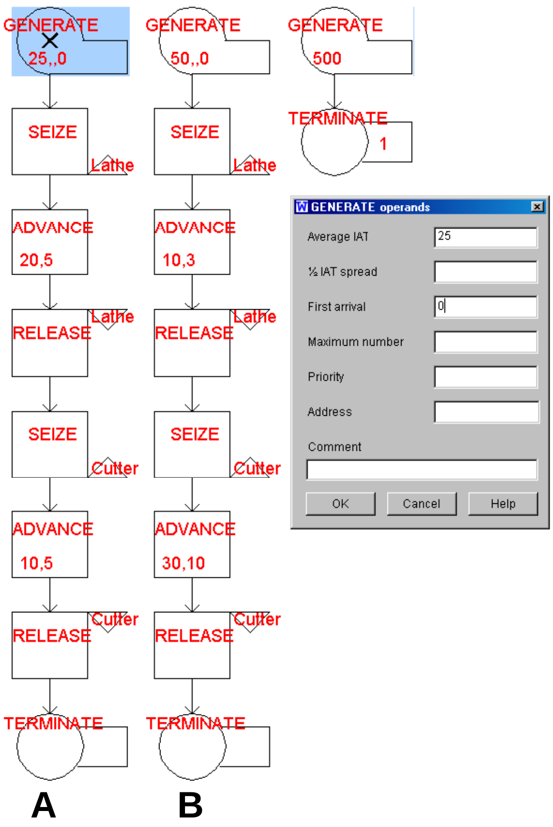


Feasibility of the Production Plan

…uses GPSS, taking random variations in machine times into account

- We want to bring unprocessed parts into the engineering shop when the parts are needed, i.e., **Just-In-Time (JIT)**.
- A JIT production policy implies the following interarrival times for parts A and B:
  - IAT_A = 500 min per day / 20 per day = 25 minutes
  - IAT_B = 500 min per day / 10 per day = 50 minutes
- We want the first part A to arrive at the start of the day (at time 0), to avoid that no production activity takes place during the first 25 minutes of the day.
- This will require looking at another operand of the GENERATE block.

Engineering Shop Block Diagram



GENERATE operands

Average IAT: 25

½ IAT spread:

First arrival: 0

Maximum number:

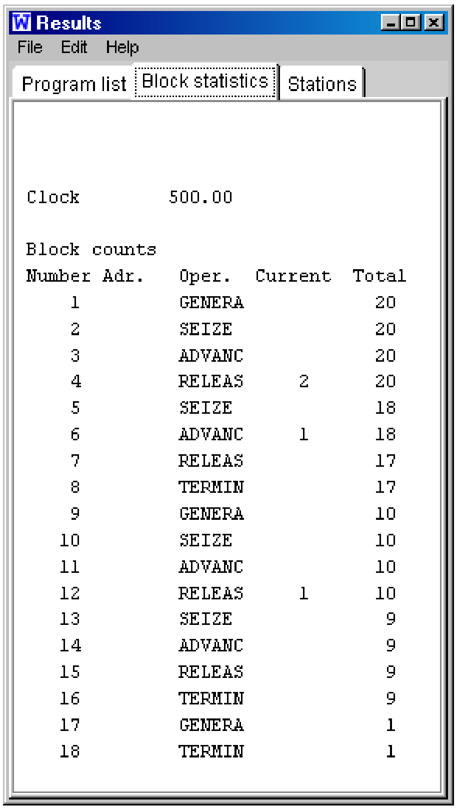
Priority:

Address:

Comment:

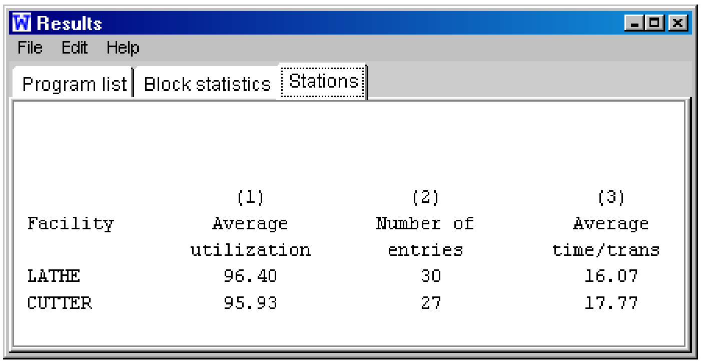
- We have two segments, one producing part A and the other producing part B.
- Each newly arriving part first uses the lathe and then the cutter.
- The third operand (the C operand) in the GENERATE operands dialog allows us to specify the time for the first arrival.
- If we do not supply a specific value, then the default value of IAT(1), the first interarrival time, would be used.
- The operands show up in the block diagram for the part A segment as 25,,0. The two commas indicate that the B operand, the ½ IAT spread, is to have its default value of 0.

Block & Stations Statistics



```
Clock          500.00

Block counts
Number Adr.      Oper.   Current   Total
     1           GENERA              20
     2           SEIZE               20
     3           ADVANC              20
     4           RELEAS     2        20
     5           SEIZE               18
     6           ADVANC     1        18
     7           RELEAS              17
     8           TERMIN              17
     9           GENERA              10
    10           SEIZE               10
    11           ADVANC              10
    12           RELEAS     1        10
    13           SEIZE                9
    14           ADVANC               9
    15           RELEAS               9
    16           TERMIN               9
    17           GENERA               1
    18           TERMIN               1
```

- **Question #1** How many part A and part B were produced?
- **Question #2** What do the current counts in blocks 4, 6,and 12 represent?
- **Question #3** Is the production plan based upon linear programming feasible?
- **Question #4** Were the lathe and cutter used at full capacity? Why or why not?

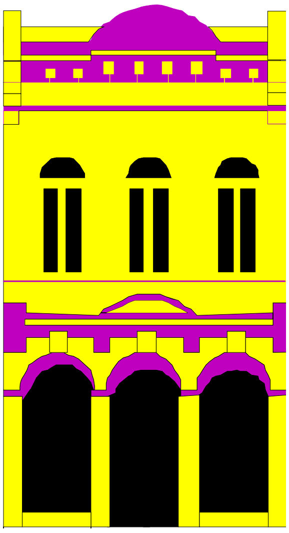


```
                    (1)            (2)            (3)
Facility          Average       Number of       Average
                utilization      entries       time/trans
LATHE             96.40            30             16.07
CUTTER            95.93            27             17.77
```

Small Store Simulation Model

- Customers arrive every 7 ± 3 minutes, randomly and uniformly distributed.
- Two people work in the store, Joe and Cash.
- Customers first go to Joe to choose goods and find out how much they have to pay.
- Next they go to Cash to pay and obtain a receipt.
- Finally, they return to Joe to pick up goods after presenting a receipt. They must go to the end of this line again, just as they did the first time Joe served them.
- The store is closed after eight hours.
- We are interested in measuring the time that customers spend in the store.

In particular, is there substantial risk (10%), that a customer has to spend more than an hour in the store?

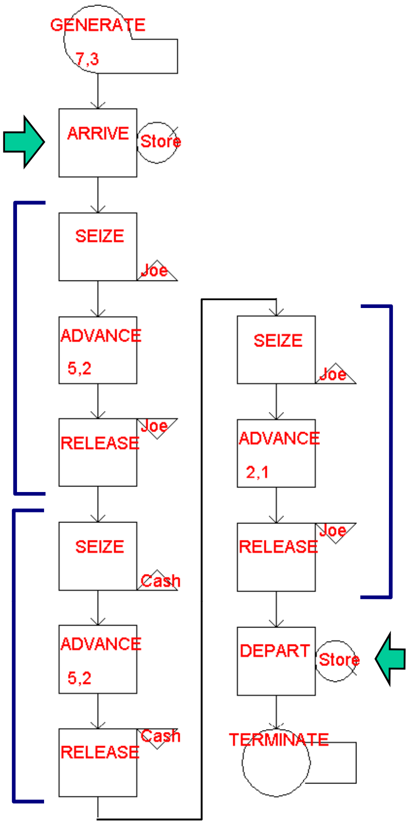


Store Block Diagram

Start measuring time spent in the store.

Customers first go to Joe to choose goods and find out how much they have to pay.

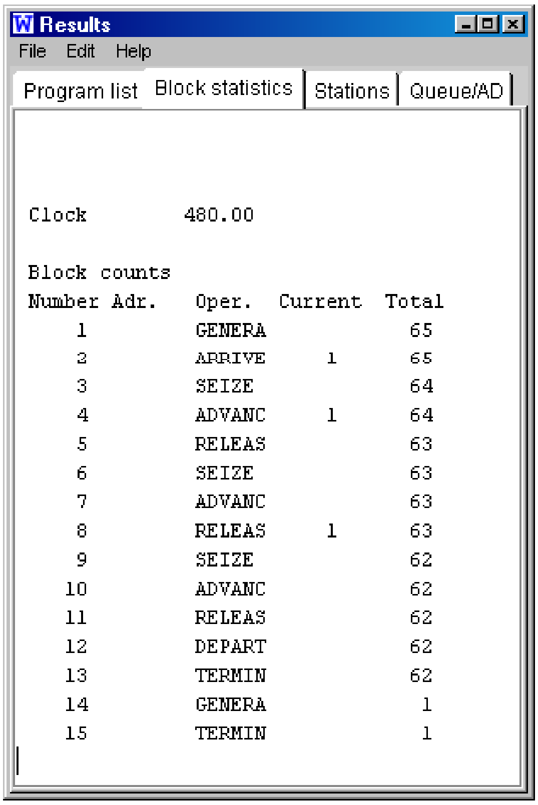
Next they go to Cash to pay and obtain a receipt.



Finally, they return to Joe to pick up goods after presenting a receipt.

Stop measuring time spent in the store.

Results – Block Statistics

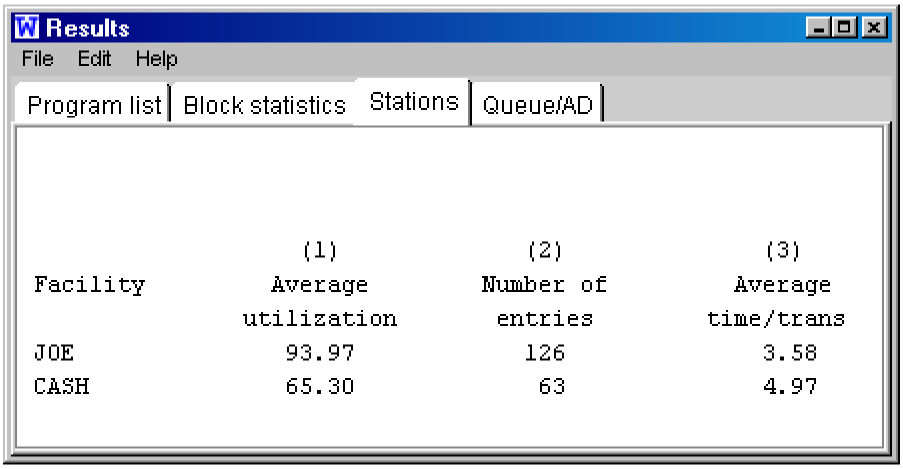


```
Clock         480.00

Block counts
Number Adr.     Oper.  Current  Total
    1           GENERA             65
    2           ARRIVE      1      65
    3           SEIZE              64
    4           ADVANC      1      64
    5           RELEAS             63
    6           SEIZE              63
    7           ADVANC             63
    8           RELEAS      1      63
    9           SEIZE              62
   10           ADVANC             62
   11           RELEAS             62
   12           DEPART             62
   13           TERMIN             62
   14           GENERA              1
   15           TERMIN              1
```

Question #1 How many customers are waiting for Joe at the end of the day?

Question #2 Is anyone in the process of paying at the end of the day? How do you determine this?

Results – Stations Statistics

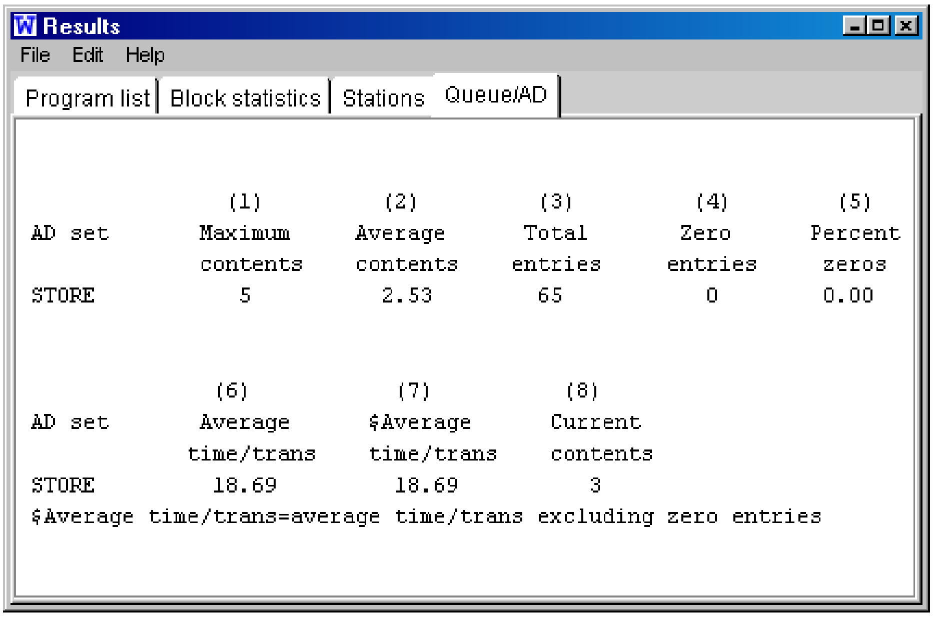


```
                    (1)            (2)             (3)
Facility          Average       Number of        Average
                utilization      entries       time/trans
JOE               93.97           126             3.58
CASH              65.30            63             4.97
```

Question #1 How do the utilization data for Joe and cash compare?

Question #2 Does the high utilization of Joe suggest a potential problem?

Results – AD Statistics



```
              (1)        (2)         (3)        (4)       (5)
AD set      Maximum    Average     Total      Zero     Percent
            contents   contents    entries    entries    zeros
STORE          5         2.53        65         0        0.00


             (6)          (7)          (8)
AD set      Average      $Average     Current
           time/trans   time/trans   contents
STORE        18.69        18.69          3
$Average time/trans=average time/trans excluding zero entries
```

Question What is the average time that customers spend in the store?

Risk Assessment

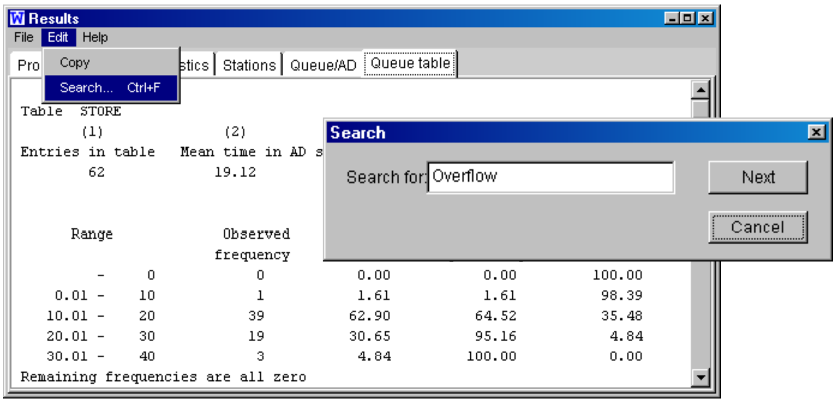
Is there substantial risk (10%), that a customer has to spend more than an hour in the store?

- To answer this question we set up a queue table of times spent in the AD set *Store:*
  - *Class 1 top* is 0 minutes.
  - Class *width* is 10 minutes.
  - *# of classes* is 7
- With 7 classes, we will get an overflow whenever we have a least one customer who spends more than 60 minutes in the store.

…0]	**(0…10]**	**(10…20]**	**(20…30]**	**(30…40]**	**(40…50]**	**(50…60]**	**minutes**
1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**	**class**

- The program is then run 10 times, so that we can see in the queue table statistics if we get an ***Overflow*** in any of the 10 runs.

Risk Assessment

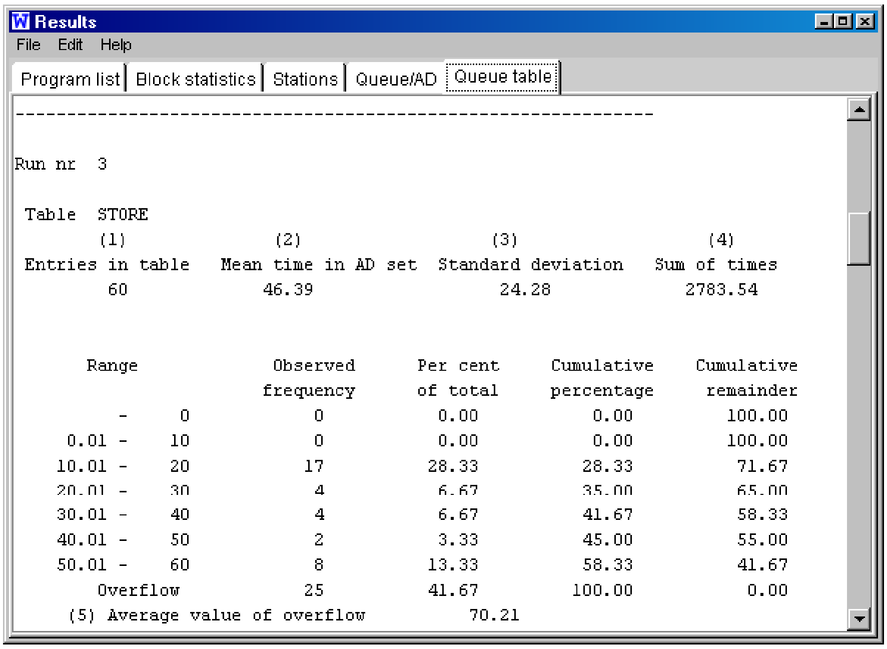


Overflow

```
Table   STORE
      (1)                (2)
Entries in table    Mean time in AD s
      62               19.12


       Range             Observed
                         frequency
           -      0          0          0.00         0.00        100.00
     0.01  -     10          1          1.61         1.61         98.39
    10.01  -     20         39         62.90        64.52         35.48
    20.01  -     30         19         30.65        95.16          4.84
    30.01  -     40          3          4.84       100.00          0.00
Remaining frequencies are all zero
```

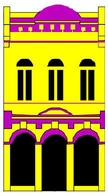
- We select the queue table tab from the Results window.
- We click on Edit|Search.
- We then search for the word *Overflow*.

Risk Assessment



```
--------------------------------------------------------------

Run nr  3

 Table  STORE
        (1)              (2)                 (3)                 (4)
 Entries in table   Mean time in AD set  Standard deviation   Sum of times
         60              46.39                24.28              2783.54



       Range               Observed      Per cent     Cumulative    Cumulative
                           frequency     of total     percentage    remainder
           -       0           0           0.00          0.00         100.00
     0.01  -      10           0           0.00          0.00         100.00
    10.01  -      20          17          28.33         28.33          71.67
    20.01  -      30           4           6.67         35.00          65.00
    30.01  -      40           4           6.67         41.67          58.33
    40.01  -      50           2           3.33         45.00          55.00
    50.01  -      60           8          13.33         58.33          41.67
        Overflow              25          41.67        100.00           0.00
     (5) Average value of overflow             70.21
```

- **Question #1** How many of the 60 customers spent more than one hour in the store?
- **Question #2** What was the average time that these very patient customers spent in the store?



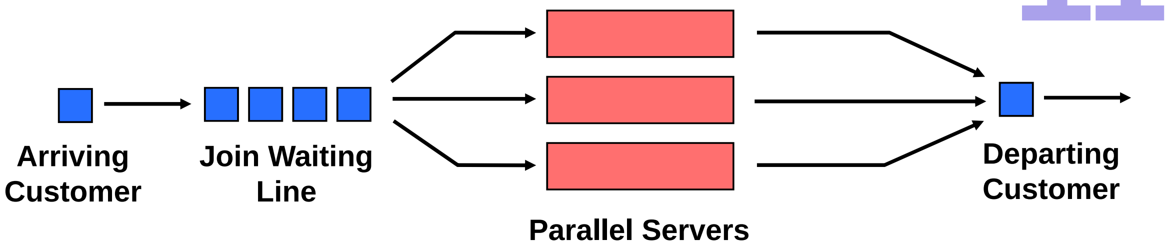
Lesson 11

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

The GPSS Storage Concept



- In all previous lessons, stations have been **facilities**, serving only one customer at a time.
- We now study stations called **storages** that can serve several customers simultaneously.
- This concept can be used when the following conditions all hold:
  - The servers work in parallel.
  - All customers wait in a single joint queue.
  - The differences between the individual servers, as concerns service times, are small.
  - The customers cannot choose a server, but go to the first one that becomes idle.

A Barbershop with Two Barbers

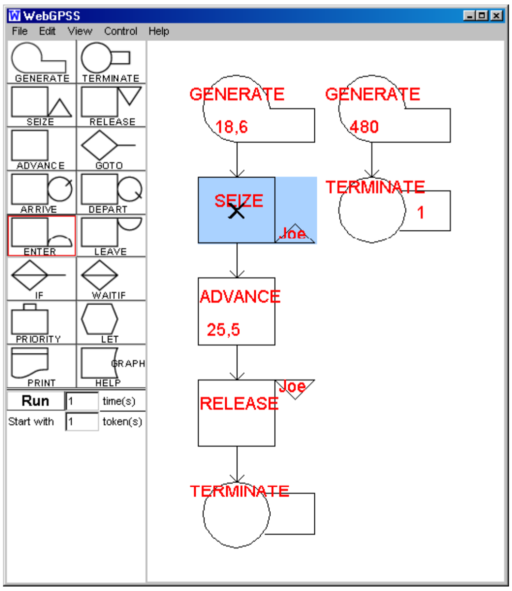
- We will modify the barbershop model of PRO07 so that we have two barbers working instead of one.
- Upon arriving, the second customer does not wait, as he can be served by the second barber.
- Instead of trying to seize a barber, a customer will *enter* the barbershop and attempt to get service from any available barber.

An ENTER block will thus replace the SEIZE block when the station is a storage.

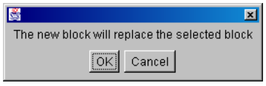
- Instead of releasing a barber, a customer will *leave* the barbershop when the haircut is done.

A LEAVE block will thus replace the RELEASE block when the station is a storage.

Replacing Blocks in Block Diagrams

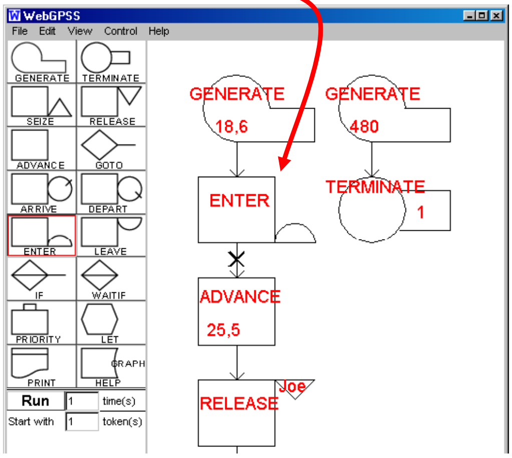


- Click on the SEIZE block to mark it in blue.
- Click on the ENTER block in the symbol menu.
- The following warning appears:



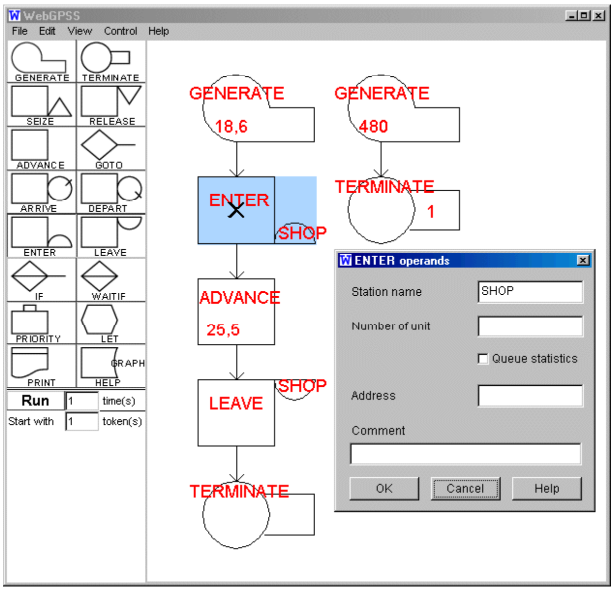
The new block will replace the selected block

- Click OK and ENTER will replace the SEIZE block.



- In a similar fashion, the RELEASE block is replaced by the LEAVE block.

ENTER and LEAVE Operands



ENTER operands

Station name: SHOP

Number of unit:

☐ Queue statistics

Address:

Comment:

OK | Cancel | Help

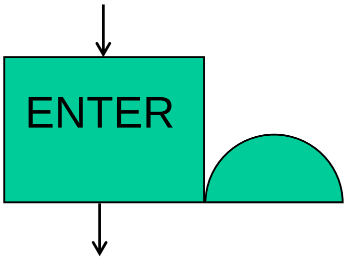
- Double-click on the ENTER block to open its operands dialog.
- Key in SHOP as the name of the station, then click OK.
- Similarly, supply the name SHOP for the LEAVE block station name.

Naming rules for storages follows the general naming rules for WebGPSS: 3 letters followed by 0-3 letters or digits.

The name of a storage must **not** have the same name as a facility in the same program.

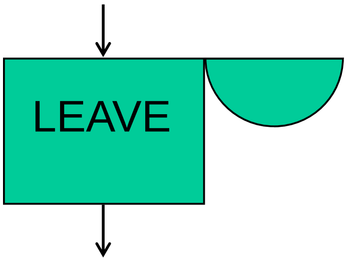
ENTER and LEAVE Blocks

- When a transaction reaches this ENTER block, it tries to enter this block.
  - If the storage is full (no servers are free) then the transaction will wait in the block that it would be in prior to ENTER. The transaction is also added to a FIFO queue
  - If the storage is not full, the transaction will go through the ENTER and on to the next block. The current contents of the storage increases by 1.



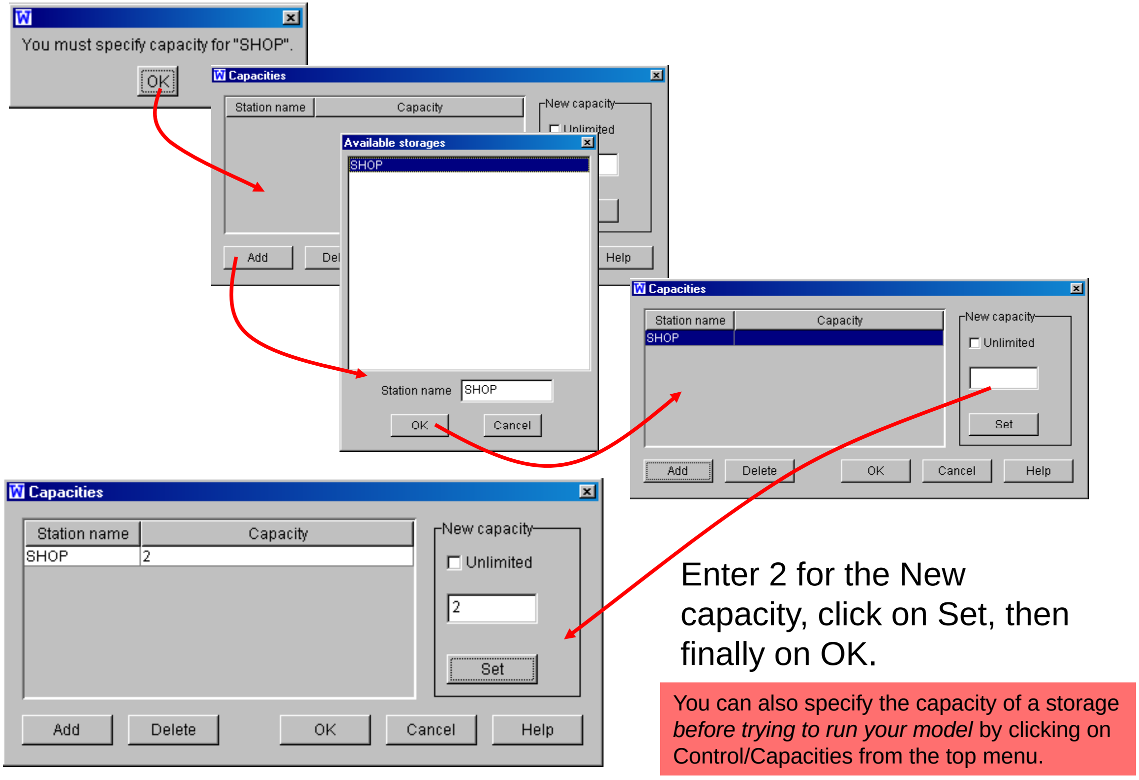
ENTER and LEAVE are mirror image pairs.

- The LEAVE block never refuses entry to a transaction. When a transaction enters this LEAVE block:
  - The storage's current contents is decreased by 1.
  - The storage's free capacity is increased by 1.
  - If the free capacity goes from 0 to 1, then the next transaction (if any) in the FIFO queue gets service.



Specifying Storage Capacity

When we try to run this model,
we get the following message:



You must specify capacity for "SHOP".

Station name: SHOP

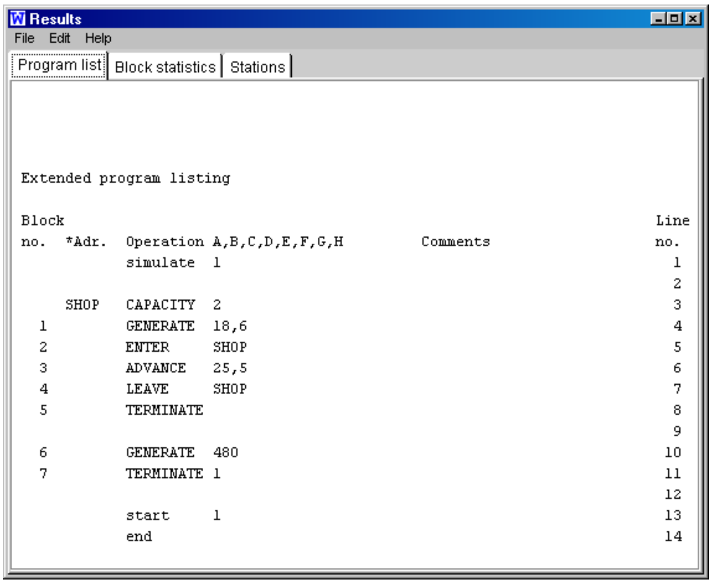
Station name	Capacity
SHOP	2

New capacity: 2

Enter 2 for the New capacity, click on Set, then finally on OK.

You can also specify the capacity of a storage *before trying to run your model* by clicking on Control/Capacities from the top menu.

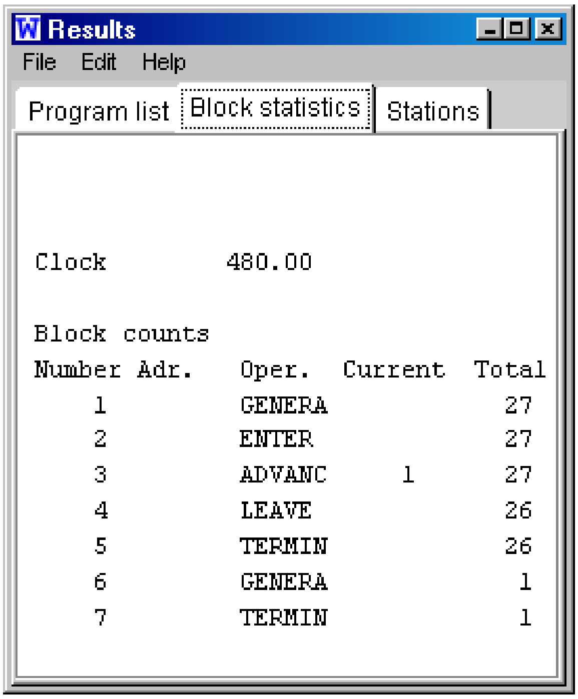
Results – Program Listing



```
Extended program listing

Block                                                        Line
no.  *Adr.  Operation A,B,C,D,E,F,G,H         Comments       no.
            simulate  1                                        1
                                                               2
     SHOP   CAPACITY  2                                        3
  1         GENERATE  18,6                                     4
  2         ENTER     SHOP                                     5
  3         ADVANCE   25,5                                     6
  4         LEAVE     SHOP                                     7
  5         TERMINATE                                          8
                                                               9
  6         GENERATE  480                                     10
  7         TERMINATE 1                                       11
                                                              12
            start     1                                       13
            end                                               14
```

- The program listing shows a CAPACITY control statement.
- The A operand is the capacity of the storage SHOP.
- SHOP has a CAPACITY of 2 units, or barbers, in this model.

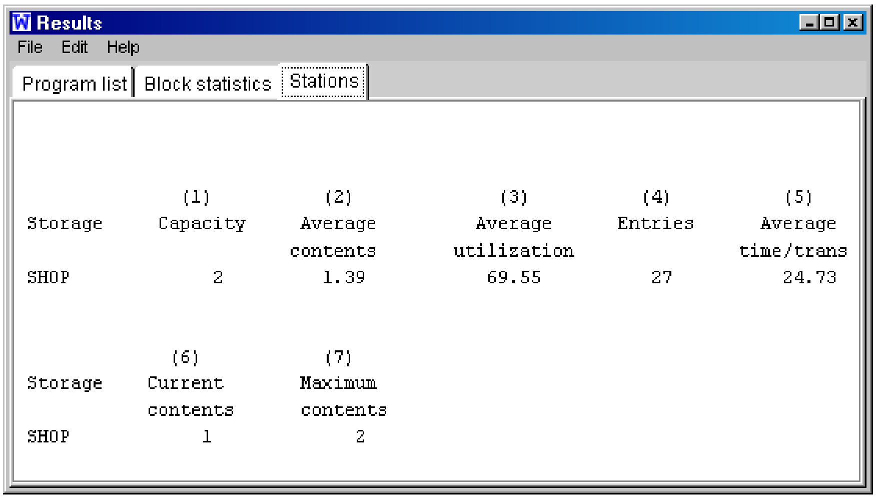
Block Statistics



```
Clock         480.00

Block counts
Number Adr.    Oper.   Current   Total
     1         GENERA               27
     2         ENTER                27
     3         ADVANC      1        27
     4         LEAVE                26
     5         TERMIN               26
     6         GENERA                1
     7         TERMIN                1
```

- **Question #1** Are there any customers waiting at closing time? How can you determine this?
- **Question #2** Is a customer getting a hair cut at closing time? How can you determine this?

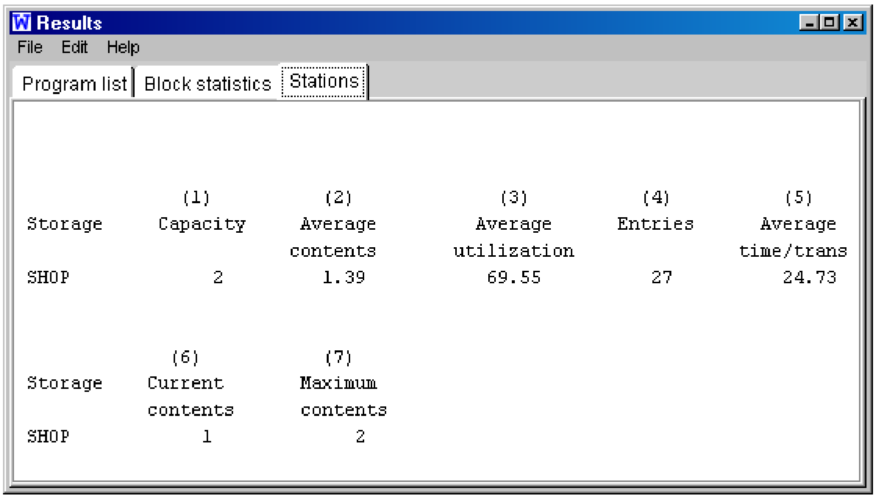
Station Statistics for a Storage



```
                (1)         (2)          (3)         (4)        (5)
Storage      Capacity     Average      Average     Entries    Average
                          contents   utilization             time/trans
SHOP             2          1.39        69.55        27        24.73


               (6)          (7)
Storage      Current      Maximum
             contents     contents
SHOP            1            2
```

1. **Capacity** The value of the capacity, or maximum number of transactions that can be served simultaneously.
2. **Average contents** The total time that all transactions have spent in the storage divided by the total simulation time.
3. **Average utilization** Expressed as a percent, this is the average contents divided by the capacity.

Station Statistics for a Storage

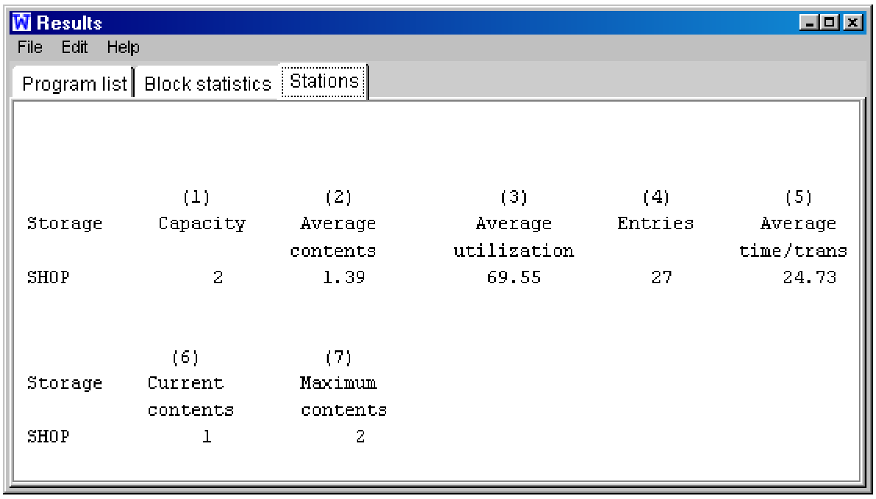


```
Storage      (1)        (2)          (3)        (4)        (5)
           Capacity   Average      Average    Entries    Average
                      contents   utilization            time/trans
SHOP          2         1.39        69.55        27        24.73


              (6)        (7)
Storage     Current    Maximum
            contents   contents
SHOP          1          2
```

4. **Entries** The total number of transactions who have started to get service, same as the total count for the ENTER block.
5. **Average time/trans** The total time spent by all transactions receiving service, divided by the number of entries.

Question Why is average time/trans generally slightly underestimated?

Station Statistics for a Storage



```
Storage      (1)        (2)          (3)        (4)        (5)
           Capacity   Average      Average    Entries    Average
                      contents   utilization            time/trans
SHOP          2         1.39        69.55        27        24.73


              (6)        (7)
Storage     Current    Maximum
            contents   contents
SHOP          1          2
```

6. **Current contents** The number of transactions who are receiving service at simulation stop time.
7. **Maximum contents** The highest number of transactions served simultaneously at any time during the simulation.

Discussion Questions

Question #1 A storage with a capacity of 1 can act similar to a facility. Why would you ever want to define a storage this way?

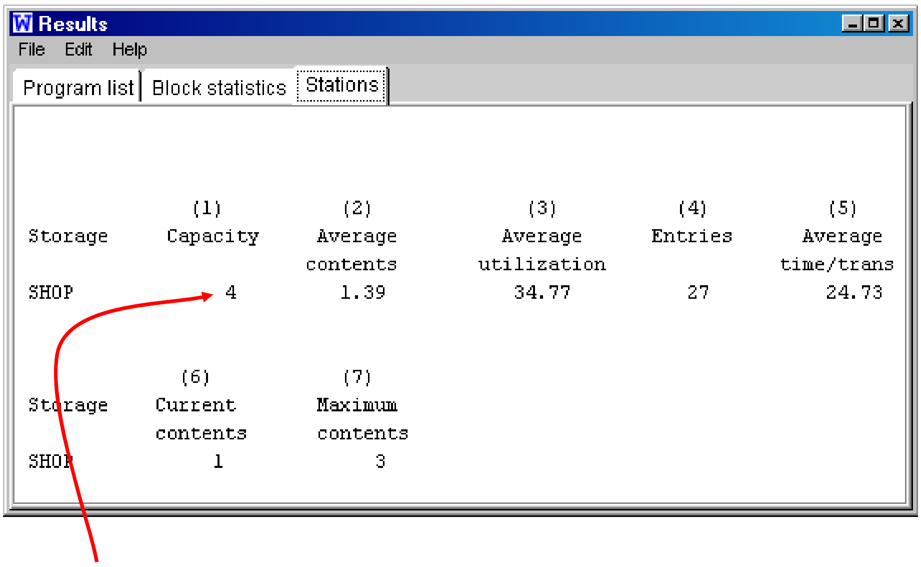
Question #2 Why might you sometimes want to set the capacity of a storage to a high number?

Answer

Question #3 How would you guess that one could go about obtaining queue statistics for a storage?

Answer

Setting Capacity to a High Value



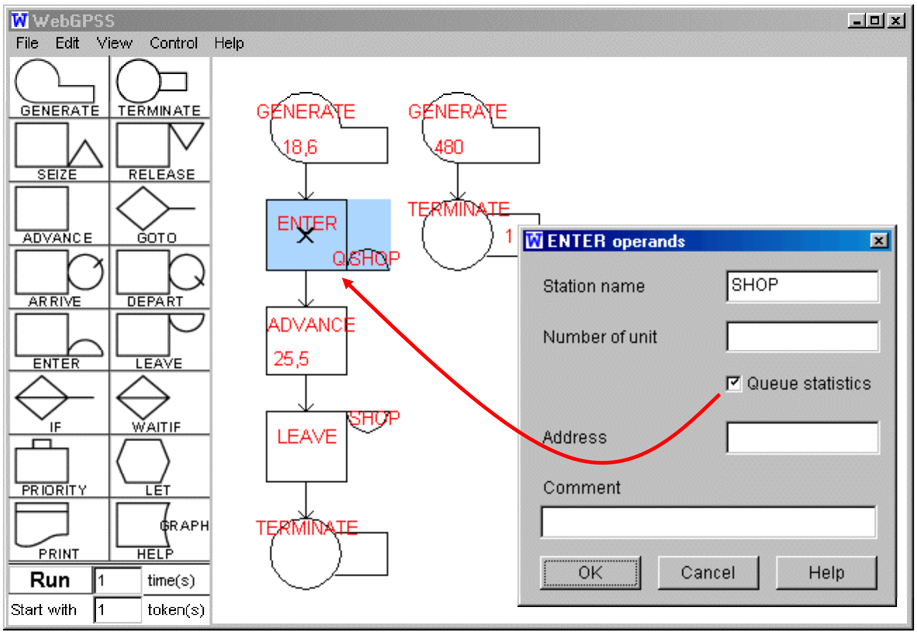
```
Storage       (1)       (2)         (3)        (4)       (5)
           Capacity   Average     Average    Entries   Average
                      contents  utilization           time/trans
SHOP           4        1.39       34.77       27        24.73


              (6)       (7)
Storage     Current   Maximum
            contents  contents
SHOP           1         3
```

Here, we modified the barbershop model only by changing the capacity of SHOP to 4 barbers.

Question What do we learn from the stations statistics?

Back

Obtaining Storage Queue Statistics



ENTER operands

Station name: SHOP

Number of unit:

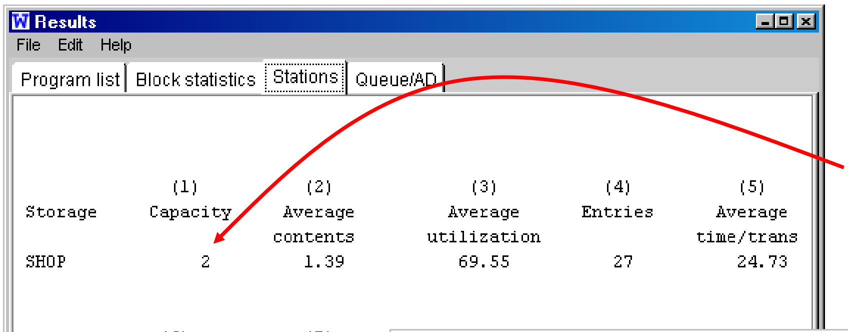
☑ Queue statistics

Address:

Comment:

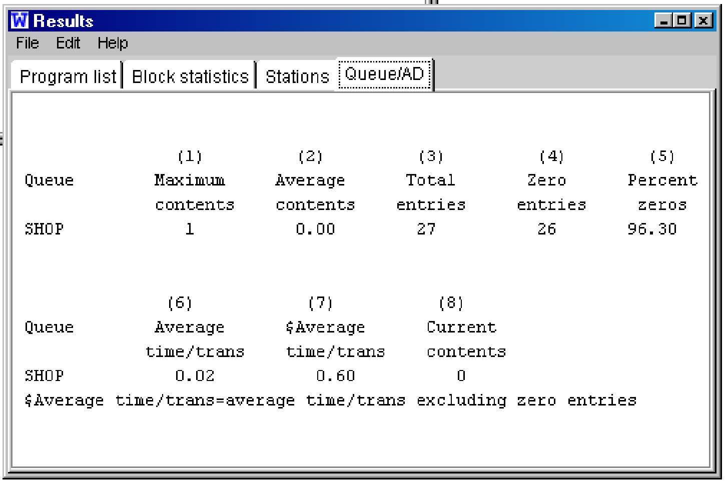
Just as we did with SEIZE, we check the *Queue statistics* box in the ENTER operands dialog.

Obtaining Storage Queue Statistics



```
Storage        (1)         (2)           (3)         (4)        (5)
            Capacity     Average       Average     Entries    Average
                         contents    utilization             time/trans
SHOP            2          1.39         69.55        27         24.73

               (6)         (7)
Storage     Current      Maximum
            contents     contents
SHOP           1            2
```

Note that we changed the number of barbers back to 2.



```
Queue        (1)        (2)        (3)        (4)        (5)
           Maximum    Average     Total      Zero      Percent
           contents   contents   entries    entries     zeros
SHOP          1         0.00       27         26        96.30

               (6)        (7)        (8)
Queue        Average   $Average    Current
            time/trans time/trans  contents
SHOP          0.02       0.60        0
$Average time/trans=average time/trans excluding zero entries
```

Question #1 How many customers had to wait?

Question #2 How many minutes was this wait?

WebGPSS

12

Lesson 12

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

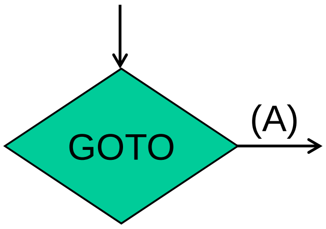
UIN Sunan Kalijaga Yogyakarta

The Simplest form of the GOTO Block

12

WebGPSS

- Up to now, transactions have moved sequentially from one block to the next.
- We now study the simplest form of the GOTO block, which allows a transaction to jump to a block that is not the next block.
- This block causes a transaction that enters the block to go next to the block with the address specified by the A operand.
- There must be a block in the GPSS program whose address field matches the address specified by the A operand of the GOTO block.
- This form of the GOTO is often referred to as an *unconditional GOTO*.



A Pottery Shop Model

- The model of this lesson concerns a pottery, where a number of potters throw large pots
- Throwing a pot takes between 25 and 35 minutes.
- The pot is then placed in an oven (kiln), which has room for only one pot at a time.
- The pot remains in the oven from 6 to 10 minutes, during which it must be watched by the potter, so that it will not crack.
- When the pot is removed from the oven, the potter starts throwing a new pot.
- Work continues in this way for a 10-hour workday.

With only one oven available, the question of concern is how many potters should be hired: 4 ,5 or 6?

Block Diagram – 4 Potters Working

The four potters arrive at the same time in the morning, at simulation clock time 0.

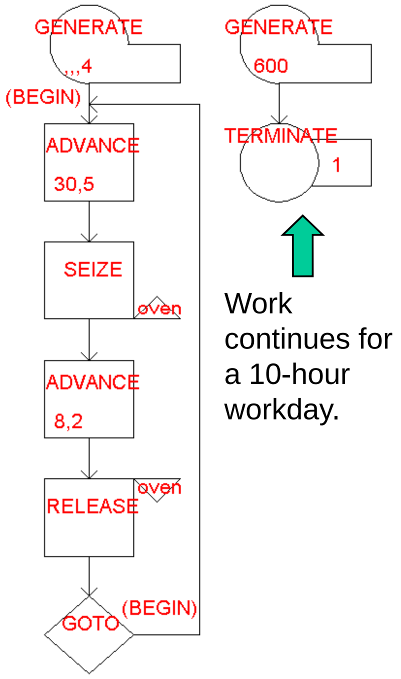
It takes a potter between 25 and 35 minutes to throw a pot.

Each potter tries to bring his pot to the oven.

The potter watches the pot while it is in the oven, so that the pot will not crack.

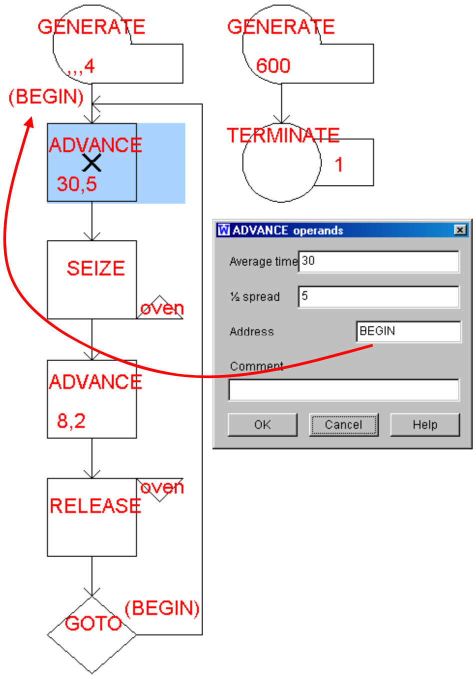
The potter frees the oven.

The potter returns to his potter's wheel to throw another pot.



Work continues for a 10-hour workday.

Specifying an Address Field



ADVANCE operands

Average time 30

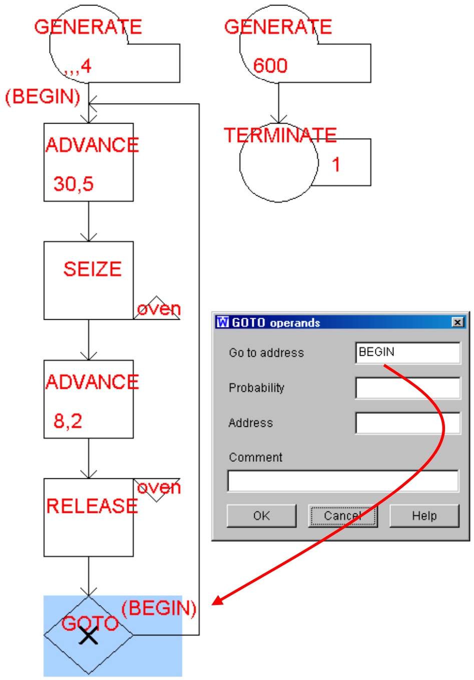
½ spread 5

Address BEGIN

Comment

- Let us first look at how the new jump construction works.
- We specify the address BEGIN in the “Address” field of the ADVANCE block.
- This will bring the potter-transactions to this ADVANCE block from the GOTO.
- Note that the address BEGIN appears, within parentheses, to the left and slightly above the ADVANCE block.

GOTO Operands Dialog



GOTO operands

Go to address BEGIN

Probability

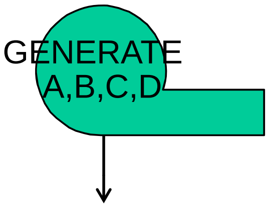
Address

Comment

- We specify BEGIN as the A operand, the *Go to address,* of the GOTO block.
- This indicates the address of the block where a potter should go after completing a pot.
- As regards the block diagram, two things should be noted:
  - *The address BEGIN is written,within parentheses, on the line extending to the right of the GOTO block.*
  - *A line with an arrow is drawn upwards from the GOTO to the first ADVANCE block.*

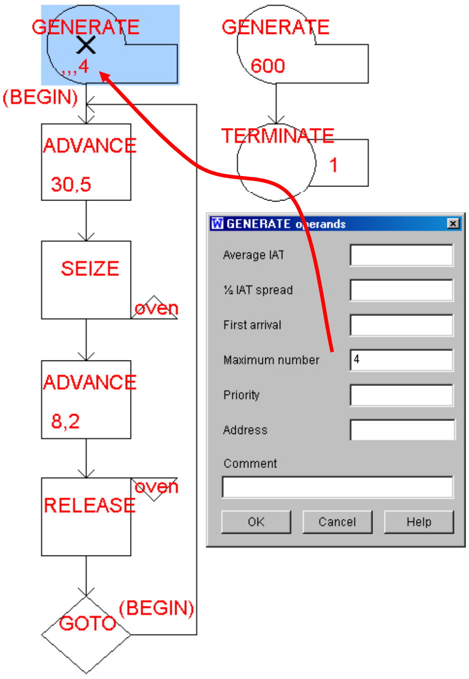
GENERATE Block Update

Operand	Description	Default
A	Average interarrival time	0
B	Half-width of the interval [A-B,A+B]	0
C	Time for arrival of the first transaction	IAT(1)
D	Maximum number of transactions the GENERATE block can create during the simulation (until the termination counter becomes zero or negative).	32,000



Question Consider the case of the block
GENERATE ,,,4.
Precisely what will this block do?

GENERATE Operands Dialog



GENERATE operands

Average IAT

½ IAT spread

First arrival

Maximum number 4

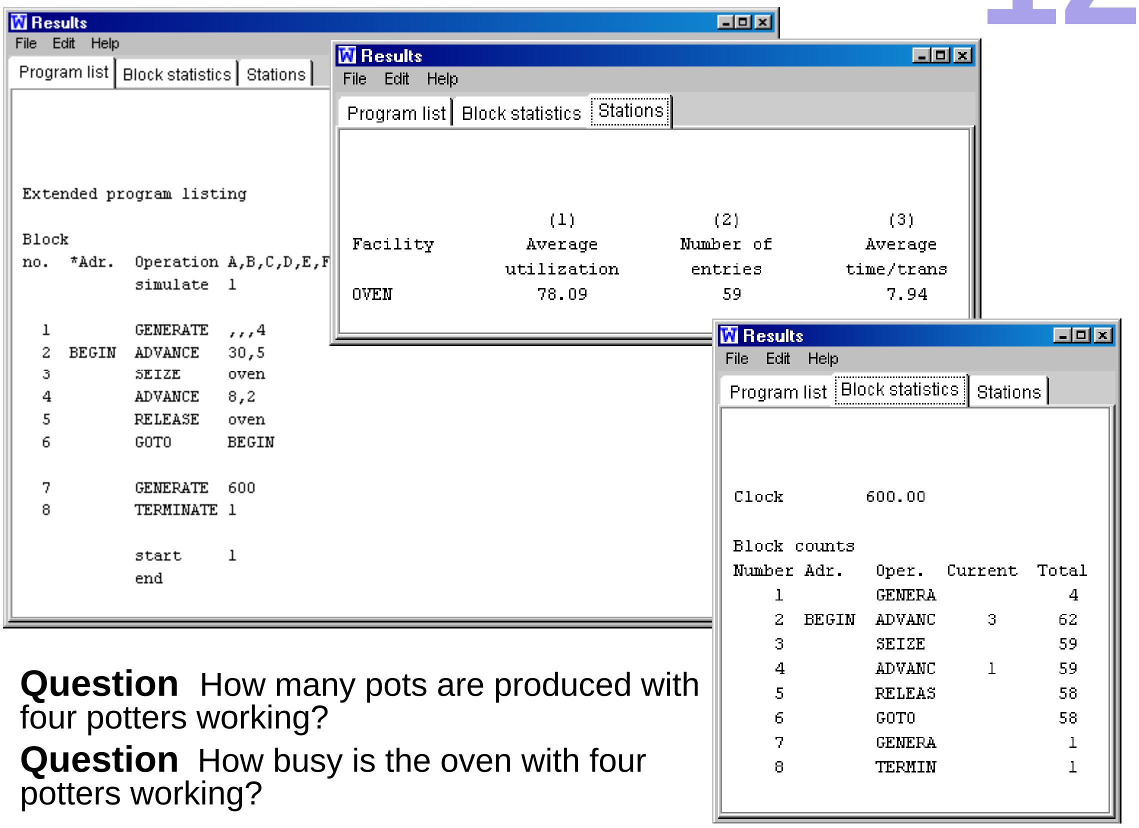
Priority

Address

Comment

- The A, B, and C operands take their default values.
- The fourth line down is for the D operand, the maximum number of transactions that can be created.
- This block will create exactly 4 potter transactions at the start of the simulation, with no other transactions being generated thereafter.

Results – 4 Potters



```
Extended program listing

Block
no.  *Adr.  Operation A,B,C,D,E,F
            simulate  1

 1          GENERATE  ,,,4
 2   BEGIN  ADVANCE   30,5
 3          SEIZE     oven
 4          ADVANCE   8,2
 5          RELEASE   oven
 6          GOTO      BEGIN

 7          GENERATE  600
 8          TERMINATE 1

            start     1
            end
```

```
                    (1)           (2)            (3)
Facility          Average      Number of       Average
                utilization     entries       time/trans
OVEN               78.09          59             7.94
```

```
Clock        600.00

Block counts
Number Adr.    Oper.   Current   Total
    1          GENERA              4
    2   BEGIN  ADVANC      3      62
    3          SEIZE              59
    4          ADVANC      1      59
    5          RELEAS             58
    6          GOTO               58
    7          GENERA              1
    8          TERMIN              1
```

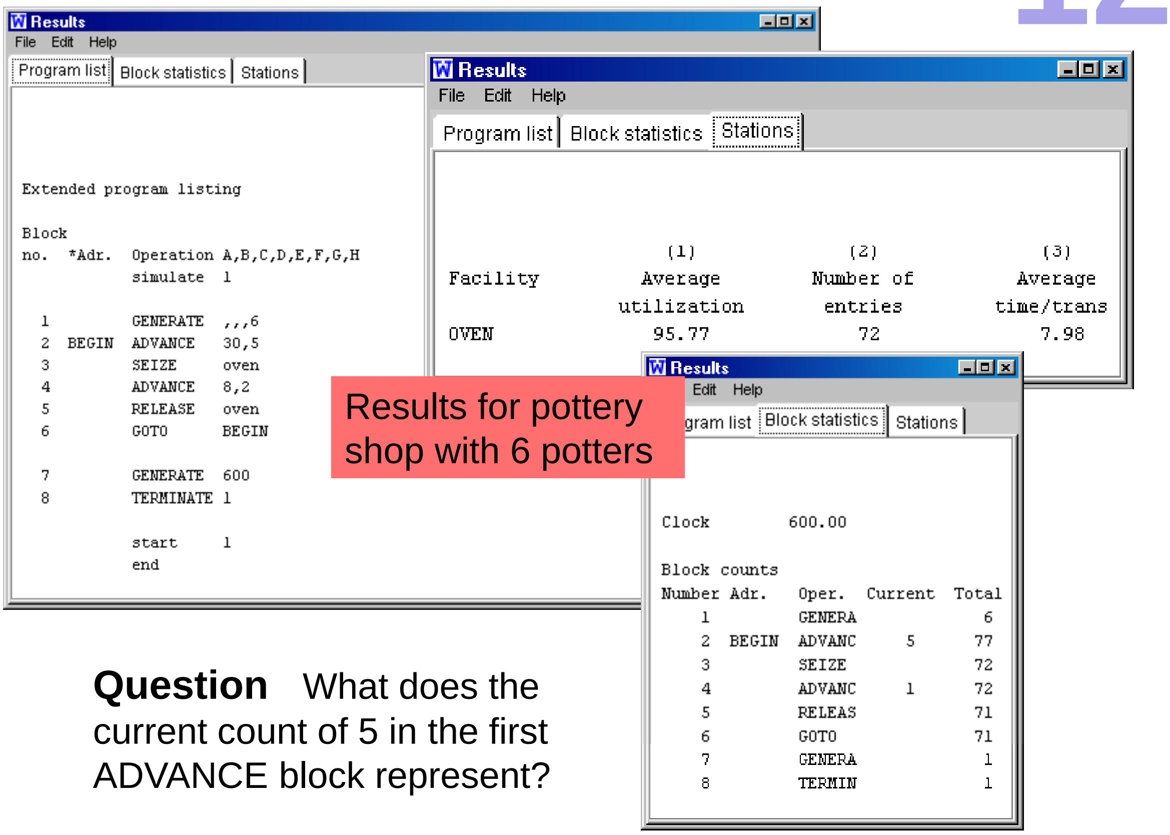
- **Question** How many pots are produced with four potters working?
- **Question** How busy is the oven with four potters working?

Results -- Summarized

Number of Potters	Pots Produced	Oven Utilization (%)
4	58	78.09
5	68	92.02
6	71	95.77

- **Question #1** What trends do you notice as the number of potters increases?
- **Question #2** Can you give some possible explanations for these trends?

A More Difficult Question



```
Extended program listing

Block
no.   *Adr.   Operation A,B,C,D,E,F,G,H
              simulate  1

 1            GENERATE  ,,,6
 2    BEGIN   ADVANCE   30,5
 3            SEIZE     oven
 4            ADVANCE   8,2
 5            RELEASE   oven
 6            GOTO      BEGIN

 7            GENERATE  600
 8            TERMINATE 1

              start     1
              end
```

```
                 (1)            (2)           (3)
Facility       Average       Number of      Average
             utilization      entries      time/trans
OVEN            95.77           72           7.98
```

```
Clock        600.00

Block counts
Number Adr.    Oper.   Current   Total
    1          GENERA              6
    2   BEGIN  ADVANC     5       77
    3          SEIZE              72
    4          ADVANC     1       72
    5          RELEAS             71
    6          GOTO               71
    7          GENERA              1
    8          TERMIN              1
```

Results for pottery shop with 6 potters

Question What does the current count of 5 in the first ADVANCE block represent?

Lesson 13

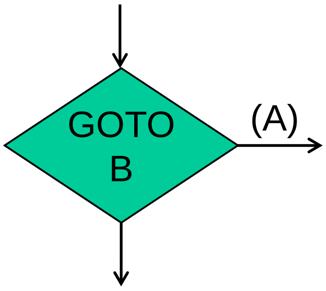
Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

The Statistical GOTO Block

- This block allows sending a transaction to another block randomly with a given probability.
- A transaction leaving this block:
  - Goes with probability B to the block with the address specified in the A operand.
  - Goes with probability 1-B to the block immediately following the statistical GOTO block.



Example: GOTO ADJUST,0.35

GPSS will sample a random uniformly distributed number between 0 and 1. If the number is less than 0.35, the transaction will go to the block whose address is ADJUST. Else, it goes to the block following the GOTO.

Quality Control Unit in a TV Factory

- TVs arrive at the control unit with an IAT between 35 and 75 seconds.
- A set is then tested by one of two inspectors during 60 to 120 seconds.
- 15% of the TVs are found defective and must be adjusted by a “fixer” who works alone, during 200 to 400 seconds.

- Adjusted TVs go back to the control unit, where they could again be found defective 15% of the time and would need to be adjusted again.
- The simulation is to be run for 80 minutes of operation.

Of specific interest is how long the TVs are delayed, waiting for inspection and for adjustment.

Quality Control Unit Block Diagram

TVs arrive at the control unit with IATs of 35-75 seconds.

We request queue statistics for the inspectors.

A TV is controlled by one of two inspectors for a time of 60 to 120 seconds. (The storage TEST must be assigned a capacity of 2.)

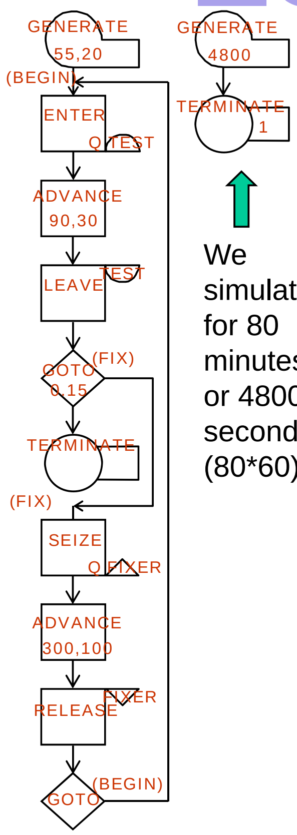
15% of the TVs have some sort of defect and must be sent to a fixer for repair.

Good TVs are ready to leave the factory.

We request queue statistics for the fixers.

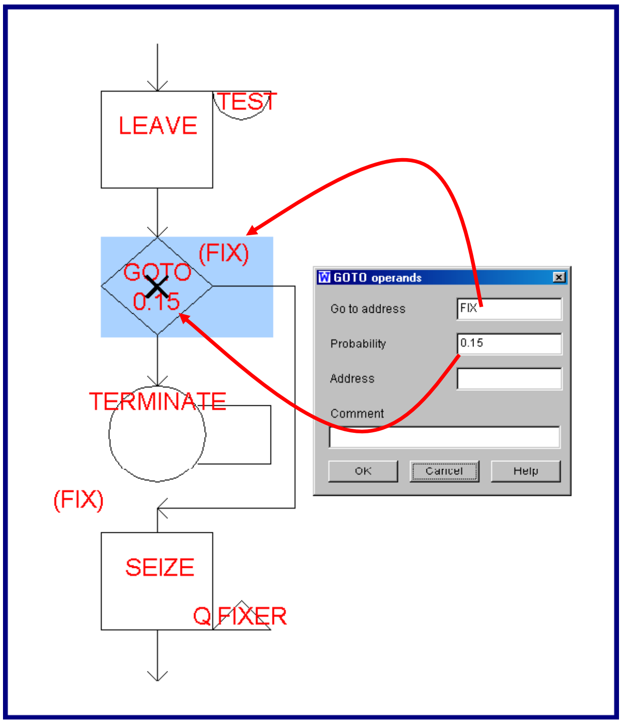
A fixer, who works alone, spends 200 to 400 seconds adjusting the defective TV.

Repaired sets go back to the control unit.



We simulate for 80 minutes, or 4800 seconds (80*60).

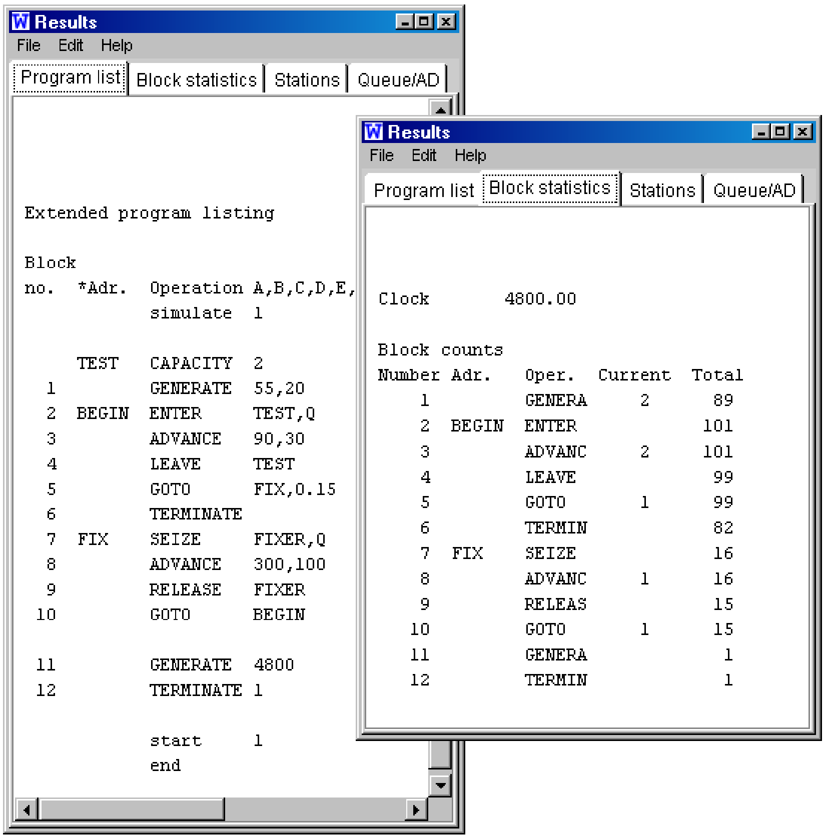
Statistical GOTO Operands Dialog



Go to address: FIX
Probability: 0.15
Address:
Comment:

- We write FIX as the *Go to address* (the A operand) of the statistical GOTO block.
- We next write 0.15 as the *Probability* (the B operand) of going to this address.

Results – Block Statistics

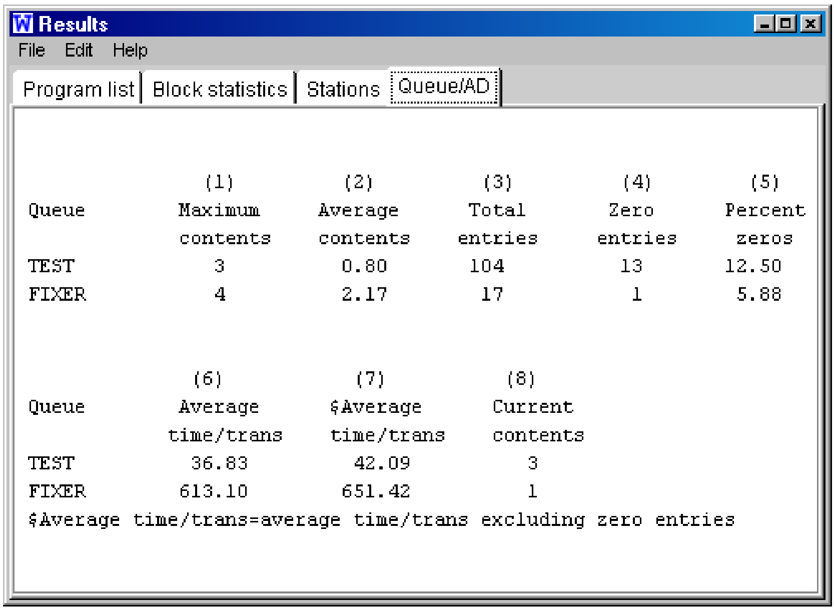


```
Extended program listing

Block
no.  *Adr.  Operation A,B,C,D,E,
            simulate  1

     TEST   CAPACITY  2
 1          GENERATE  55,20
 2   BEGIN  ENTER     TEST,Q
 3          ADVANCE   90,30
 4          LEAVE     TEST
 5          GOTO      FIX,0.15
 6          TERMINATE
 7   FIX    SEIZE     FIXER,Q
 8          ADVANCE   300,100
 9          RELEASE   FIXER
10          GOTO      BEGIN

11          GENERATE  4800
12          TERMINATE 1

            start     1
            end
```

```
Clock       4800.00

Block counts
Number Adr.   Oper.  Current  Total
    1         GENERA     2      89
    2  BEGIN  ENTER            101
    3         ADVANC     2     101
    4         LEAVE             99
    5         GOTO       1      99
    6         TERMIN            82
    7  FIX    SEIZE             16
    8         ADVANC     1      16
    9         RELEAS            15
   10         GOTO       1      15
   11         GENERA             1
   12         TERMIN             1
```

- **Question #1** How many TVs are being tested by an inspector when the simulation ended?
- **Question #2** How many TVs are in the process of being repaired?
- **Question #3** How many good TVs left the factory?
- **Question #4** What is the meaning of the current count of 1 in the statistical GOTO (block 5)?
- **Question #5** How many TVs are waiting to be tested by an inspector?

Results – Queue Statistics



```
                (1)        (2)        (3)        (4)       (5)
Queue         Maximum    Average     Total      Zero     Percent
              contents   contents   entries    entries    zeros
TEST             3         0.80       104        13       12.50
FIXER            4         2.17        17         1        5.88


               (6)         (7)          (8)
Queue        Average     $Average     Current
            time/trans  time/trans    contents
TEST          36.83        42.09         3
FIXER        613.10       651.42         1
$Average time/trans=average time/trans excluding zero entries
```

Question #1 How do waiting times compare for TEST and FIXER?

Question #2 What does this suggest regarding a possible improvement for this quality control operation?

Lesson 14

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

Standard Numerical Attribute (SNA)

- A variable that holds the value of a specific system attribute.
- If referenced by a transaction entering a block, the SNA contains the current value of that attribute.
- Each SNA has a different symbolic name. Only one type of SNA will be studied in this lesson—the length of a queue or the current contents of an AD set, given the symbolic name Q.
- Q is followed by a $-sign, which is, in turn, followed by the name of the queue.

Example: Q$Joe

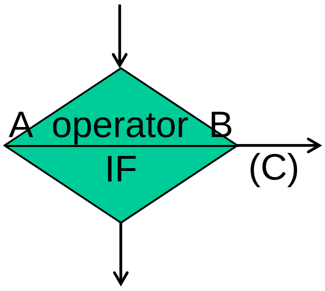
If Joe is a facility, then Q$Joe is the current length of the waiting line in front of the facility Joe.

Example: Q$TIME

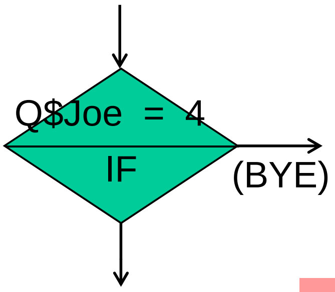
If TIME is an AD set, then Q$TIME is the number of transactions that have entered the block ARRIVE TIME but have not yet entered the block DEPART TIME, i.e., the current contents of the AD set TIME.

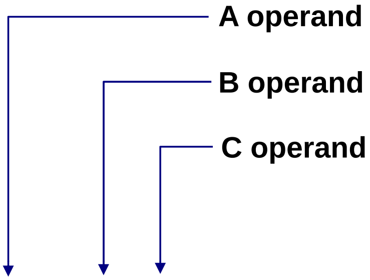
The SNA Type of IF Block

- Allows transactions to follow different paths depending upon certain conditions.
- Consists of a test condition and the address of the block to which the transaction goes if the condition is true.
- The test condition is *A operator B*.
- The C operand is the address to which the transaction in the IF block goes if the test condition is true. Else, the transaction goes to the block immediately following the IF block.
- Operator can be any one of the following six relationships:
  - = equal to
  - > greater than
  - < less than
  - >= greater than or equal to
  - <= less than or equal to
  - <> not equal to



Example SNA Type IF Block

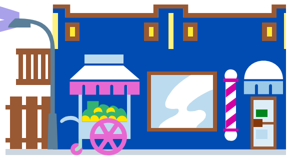




Example: IF Q$Joe=4,BYE

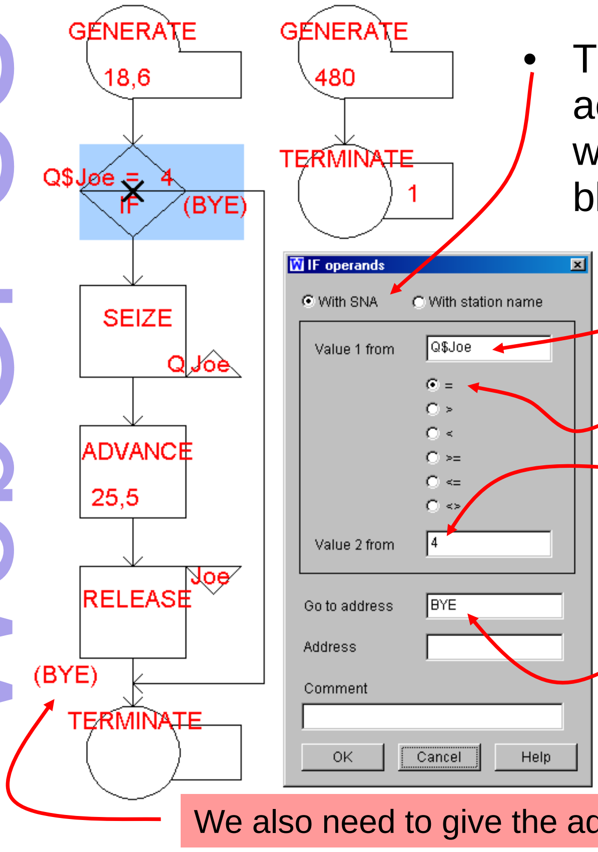
If the length of the waiting line in front of Joe is 4, then the transaction in the IF block will go to the block whose address is BYE. Else, it will go to the block following the IF block.

The Lesson 14 Simulation Model



- We will be making a modification to Joe's barbershop model of Lesson 6.
- Customers arrive randomly, between 12 and 24 minutes apart, and the time for a haircut varies between 20 and 30 minutes, with uniform distributions on these intervals.
- The barbershop closes and the simulation stops after 8 hours of operation.
- You will recall that a rather long queue developed in the Lesson 6 model.
- We now assume that customers will not want to enter and wait if the the four chairs for waiting customers are taken. In this case the customers will **balk** if there are 4 persons in the queue.

The IF Operands Dialog With SNA



● With SNA ○ With station name

Value 1 from: Q$Joe

● = ○ > ○ < ○ >= ○ <= ○ <>

Value 2 from: 4

Go to address: BYE

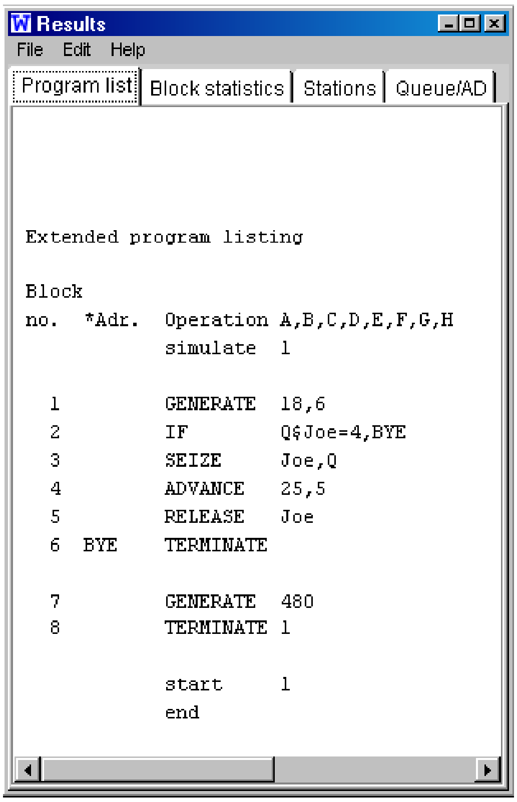
Address:

Comment:

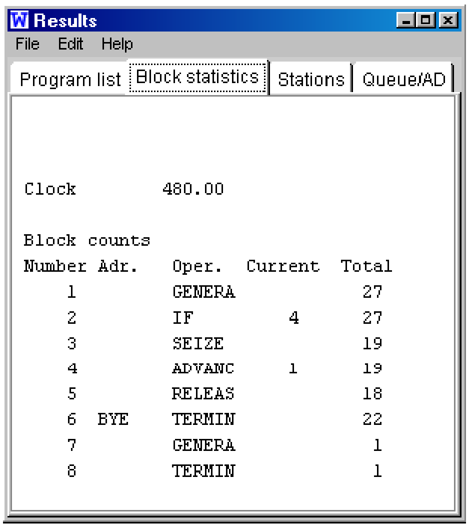
- The radio button *With SNA* is active by default. We leave it this way since this is an SNA type IF block.
  - Specify the A operand, *Value 1 from*, as Q$Joe.
  - Select = as the operator.
  - Specify the number 4 as the B operand, *Value 2 from*.
  - Specify the C operand, the *Go to address,* as BYE.

We also need to give the address BYE to the TERMINATE block.

Results – Block Statistics



```
Extended program listing

Block
no.  *Adr.  Operation A,B,C,D,E,F,G,H
            simulate  1

  1         GENERATE  18,6
  2         IF        Q$Joe=4,BYE
  3         SEIZE     Joe,Q
  4         ADVANCE   25,5
  5         RELEASE   Joe
  6  BYE    TERMINATE

  7         GENERATE  480
  8         TERMINATE 1

            start     1
            end
```

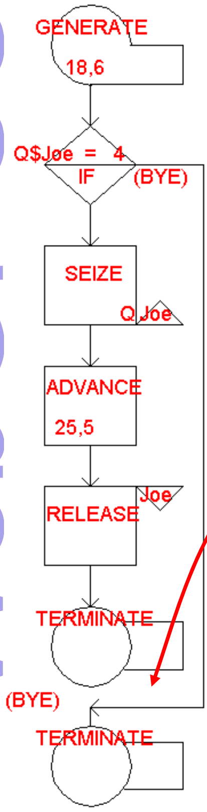


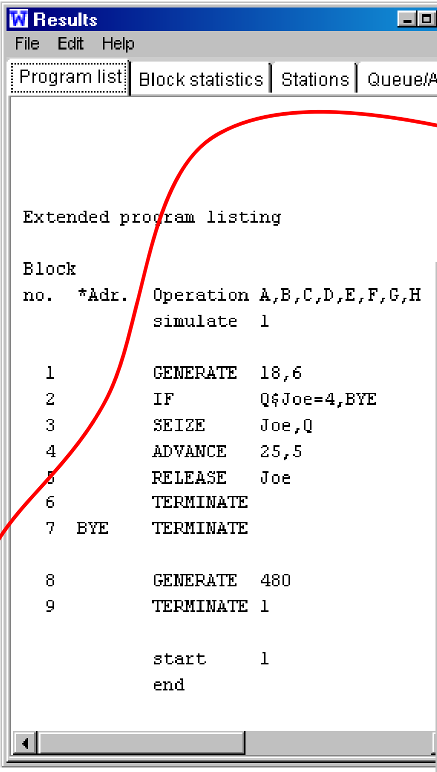
```
Clock        480.00

Block counts
Number Adr.   Oper.   Current  Total
    1         GENERA             27
    2         IF        4        27
    3         SEIZE              19
    4         ADVANC    1        19
    5         RELEAS             18
    6   BYE   TERMIN             22
    7         GENERA              1
    8         TERMIN              1
```

- **Question #1** How many customers have completed their haircuts?
- **Question #2** How many customers have balked?
- **Question #3** How could we alter the model so that we can *directly* determine the number balking? Answer

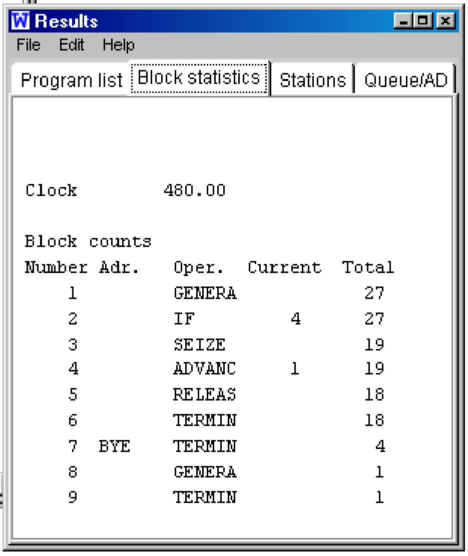
Answer to Question #3

Have separate TERMINATE blocks for balkers and for those who complete their haircuts.





```
Extended program listing

Block
no.  *Adr.  Operation A,B,C,D,E,F,G,H
            simulate  1

  1         GENERATE  18,6
  2         IF        Q$Joe=4,BYE
  3         SEIZE     Joe,Q
  4         ADVANCE   25,5
  5         RELEASE   Joe
  6         TERMINATE
  7  BYE    TERMINATE

  8         GENERATE  480
  9         TERMINATE 1

            start     1
            end
```



```
Clock        480.00

Block counts
Number Adr.   Oper.   Current  Total
    1         GENERA             27
    2         IF        4        27
    3         SEIZE              19
    4         ADVANC    1        19
    5         RELEAS             18
    6         TERMIN             18
    7   BYE   TERMIN              4
    8         GENERA              1
    9         TERMIN              1
```

WebGPSS

15

Lesson 15

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

WebGPSS

15

The Lesson 15 Simulation Model

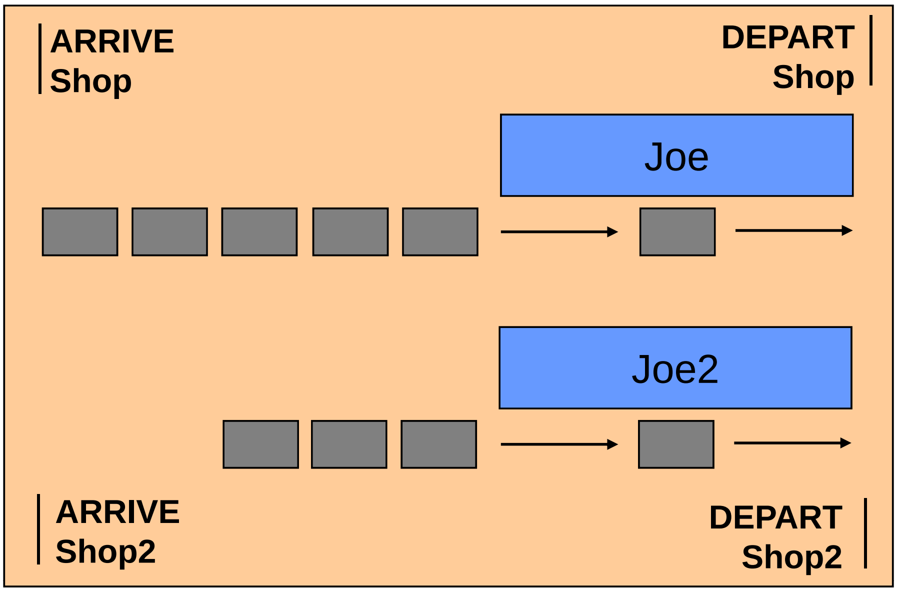
- We model a supermarket with two checkout counters, working in parallel.
- As is generally the case for supermarkets, separate waiting lines form for each counter, as each customer selects the counter he or she wishes to use.

Question #1 What factors affect which checkout counter *you* choose when you visit the supermarket?

- The checkout counters are assumed to be equivalent as regards service times. **Question #2:** How does this assumption affect your answer to question #1?

Question #3 Why can't the GPSS storage concept be used to represent the checkout counters?

The Lesson 15 Simulation Model



ARRIVE Shop

DEPART Shop

Joe

Joe2

ARRIVE Shop2

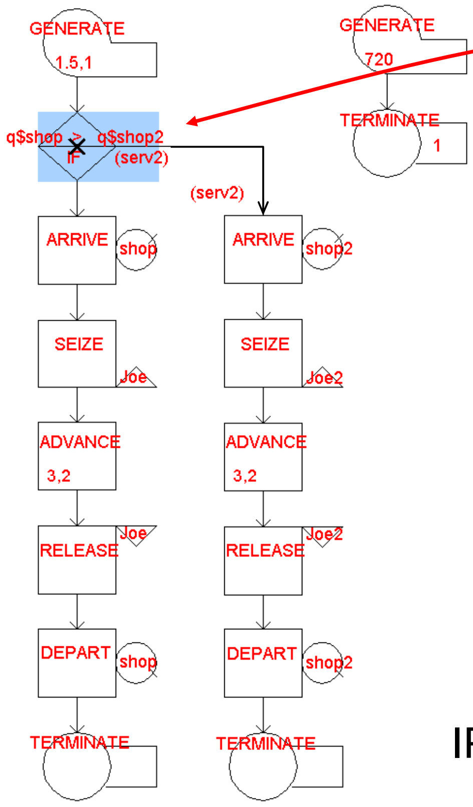
DEPART Shop2

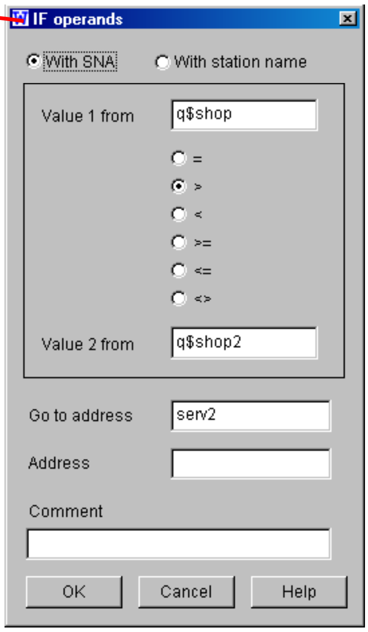
- **Question #1** What are the values of the following, as shown in the above diagram? Q$Joe, Q$Shop, Q$Joe2, Q$Shop2
- **Question #2** True or False? Q$Shop>Q$Shop2

The Lesson 15 Simulation Model

- Customers arrive at the checkout area with IATs between 0.5 minutes and 2.5 minutes, uniformly distributed on this interval.
- Service times vary uniformly between 1 and 5 minutes.
- Customers go to the checkout counter with the fewest number of people, i.e., with the smallest AD set.
- If both counters have the same number of customers, then the newly arriving customer will go to Joe (not to Joe2).
- The simulation will be run for one complete 12-hour day of operation.

Setting Up the IF Block





● With SNA ○ With station name

Value 1 from: q$shop

○ =
● >
○ <
○ >=
○ <=
○ <>

Value 2 from: q$shop2

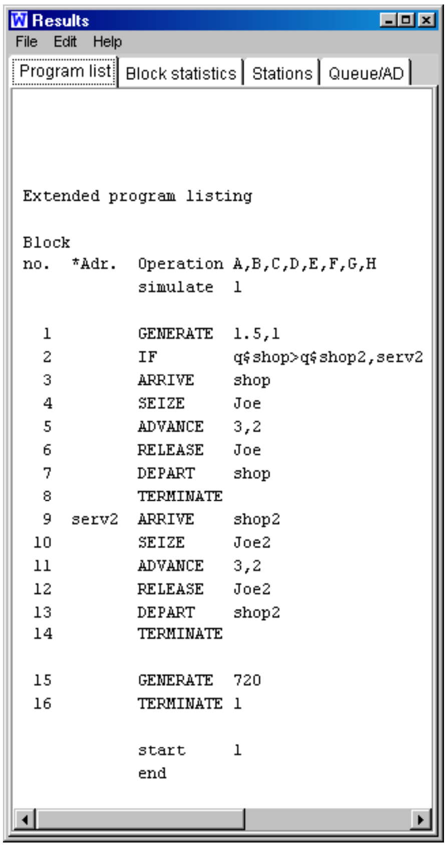
Go to address: serv2

Address:

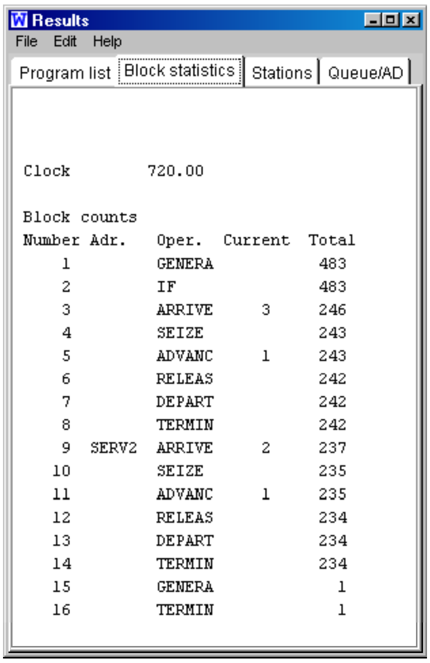
Comment:

IF q$shop>q$shop2,serv2

Results – Block Statistics



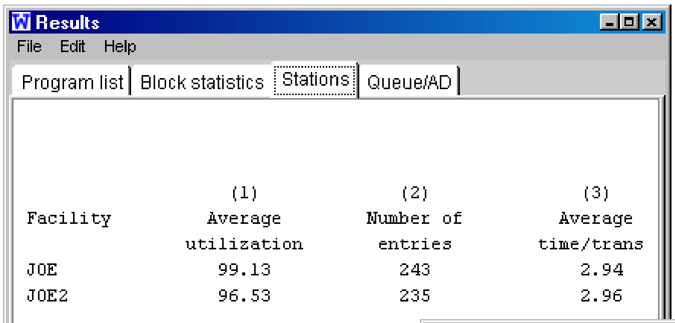
```
Extended program listing

Block
no.   *Adr.  Operation A,B,C,D,E,F,G,H
             simulate  1

 1           GENERATE  1.5,1
 2           IF        q$shop>q$shop2,serv2
 3           ARRIVE    shop
 4           SEIZE     Joe
 5           ADVANCE   3,2
 6           RELEASE   Joe
 7           DEPART    shop
 8           TERMINATE
 9    serv2  ARRIVE    shop2
10           SEIZE     Joe2
11           ADVANCE   3,2
12           RELEASE   Joe2
13           DEPART    shop2
14           TERMINATE

15           GENERATE  720
16           TERMINATE 1

             start     1
             end
```



```
Clock        720.00

Block counts
Number Adr.     Oper.  Current  Total
   1            GENERA           483
   2            IF               483
   3            ARRIVE    3      246
   4            SEIZE            243
   5            ADVANC    1      243
   6            RELEAS           242
   7            DEPART           242
   8            TERMIN           242
   9   SERV2    ARRIVE    2      237
  10            SEIZE            235
  11            ADVANC    1      235
  12            RELEAS           234
  13            DEPART           234
  14            TERMIN           234
  15            GENERA             1
  16            TERMIN             1
```

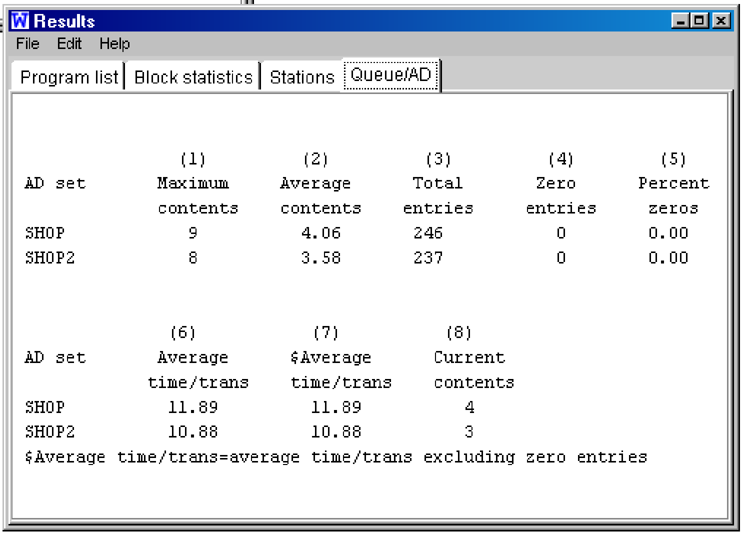
Question #1 Which checkout counters are busy and how can you tell?
Question #2 How many are waiting for service at each checkout counter?

Results – Stations & Queue/AD Statistics



```
                    (1)            (2)              (3)
Facility          Average        Number of        Average
                utilization       entries        time/trans
JOE               99.13            243              2.94
JOE2              96.53            235              2.96
```

Question #1 Why is the utilization of Joe greater than that of Joe2?



```
                (1)        (2)        (3)        (4)       (5)
AD set        Maximum    Average     Total      Zero     Percent
              contents   contents   entries    entries    zeros
SHOP             9         4.06       246         0        0.00
SHOP2            8         3.58       237         0        0.00


               (6)         (7)          (8)
AD set       Average     $Average     Current
            time/trans  time/trans    contents
SHOP          11.89       11.89          4
SHOP2         10.88       10.88          3
$Average time/trans=average time/trans excluding zero entries
```

Question #2 What is largest number of customers at Joe's counter?

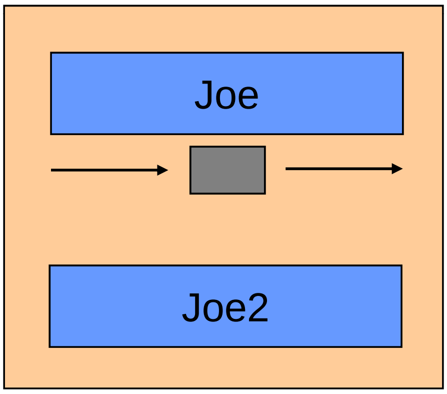
Question #3 Why are the *Average time/trans* and *$Average time/trans* equal?

A Final Question

Question What would be wrong with making use of the following IF block?

IF q$Joe>q$Joe2,serv2

Hint: Consider what the next customer arriving at the checkout area would do when encountering this situation …



Lesson 16

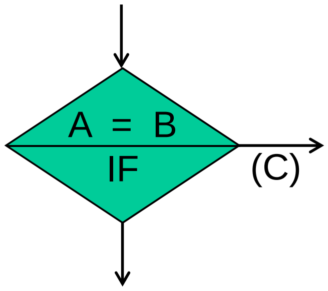
Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

UIN Sunan Kalijaga Yogyakarta

The Server Mode IF Block

- Allows transactions to follow different paths depending upon a condition concerning the name of a station (i.e., server).
- Consists of an equality test condition, A=B, and the address of the block to which the transaction goes if the condition is true.
- The A operand is the *name* of a facility or a storage.
- The B operand is a code of one or two letters, dependent on whether the A operand is a facility or a storage:



If A is a facility, then B can be one of:

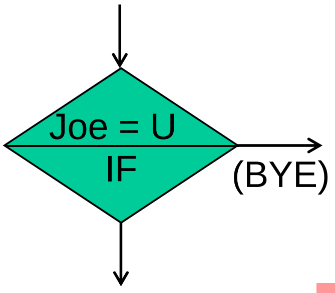
U	Facility in **U**se
NU	Facility **N**ot in **U**se

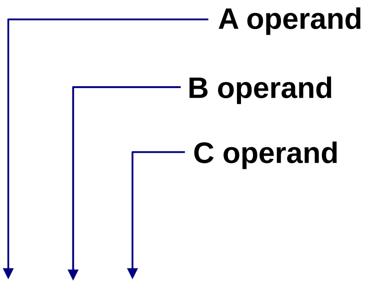
If A is a storage, then B can be one of:

E	Storage **E**mpty
NE	Storage **N**ot **E**mpty
F	Storage **F**ull
NF	Storage **N**ot **F**ull

- The C operand is the address of the block where the transaction goes if the condition is true.

Example Server Mode IF Block



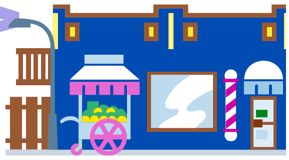


Example: IF Joe=U,BYE

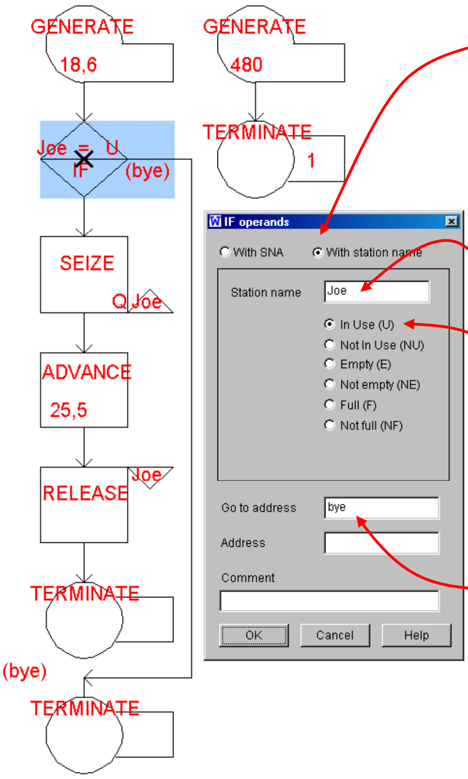
If the server Joe is in use (busy), then the transaction in the IF block will go to the block whose address is BYE. Else, it will go to the block following the IF block.

The Lesson 16 Simulation Model

- We will be making a modification to Joe’s barbershop model of Lesson 14.
- We now assume that **no** customers want to wait if the barber is busy.
- As in our previous barbershop model:
  - Customers arrive randomly, between 12 and 24 minutes apart, and the time for a haircut varies between 20 and 30 minutes, with uniform distributions on these intervals.
  - The barbershop closes and the simulation stops after 8 hours of operation.



The IF Operands Dialog With Station Name



○ With SNA ● With station name

Station name: Joe

● In Use (U)
○ Not In Use (NU)
○ Empty (E)
○ Not empty (NE)
○ Full (F)
○ Not full (NF)

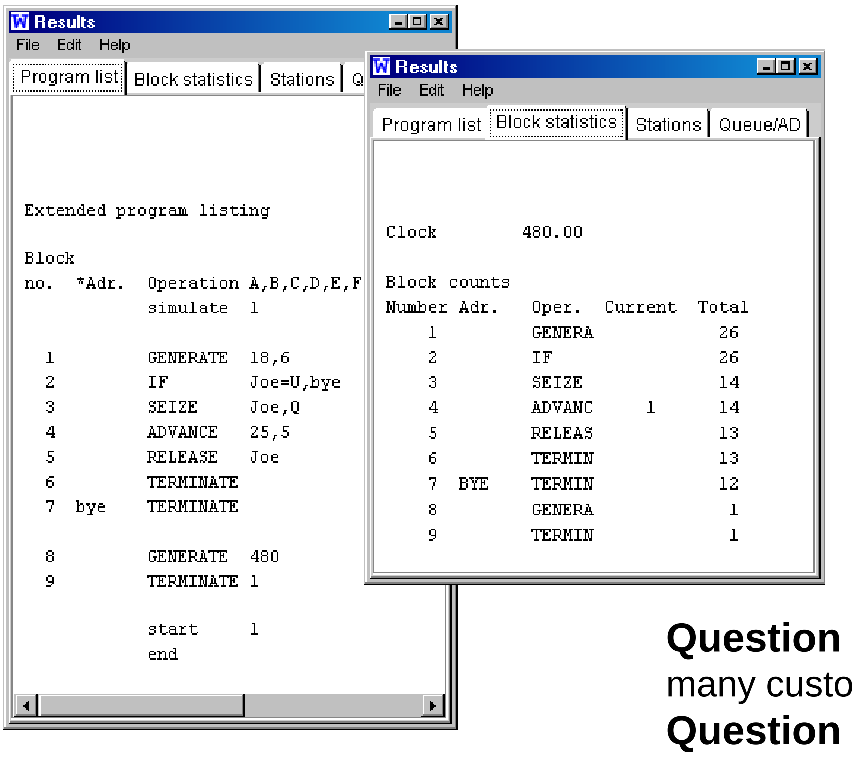
Go to address: bye

Address:

Comment:

- By default the radio button *With SNA* is active. We click on the radio button *With station name* to indicate that this will be a server mode IF block.
- Specify Joe as the A operand, *Station name*.
- Click on the radio button *In Use (U)* to indicate that the B operand is U.
- Specify the C operand, the *Go to address,* as BYE.

Results – Block Statistics



```
Extended program listing

Block
no.  *Adr.  Operation A,B,C,D,E,F
            simulate  1

  1         GENERATE  18,6
  2         IF        Joe=U,bye
  3         SEIZE     Joe,Q
  4         ADVANCE   25,5
  5         RELEASE   Joe
  6         TERMINATE
  7  bye    TERMINATE

  8         GENERATE  480
  9         TERMINATE 1

            start     1
            end
```

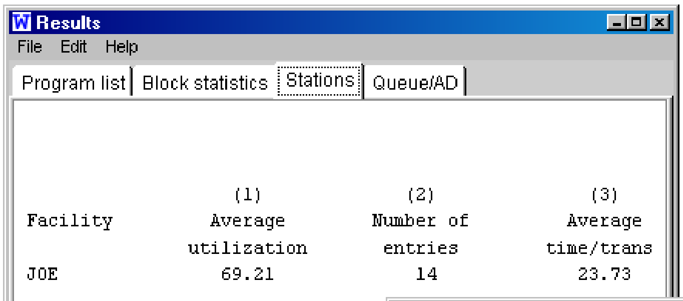
```
Clock        480.00

Block counts
Number Adr.   Oper.  Current  Total
    1         GENERA            26
    2         IF                26
    3         SEIZE             14
    4         ADVANC     1      14
    5         RELEAS            13
    6         TERMIN            13
    7  BYE    TERMIN            12
    8         GENERA             1
    9         TERMIN             1
```

Question #1 How many customers balk?

Question #2 Why have they balked?

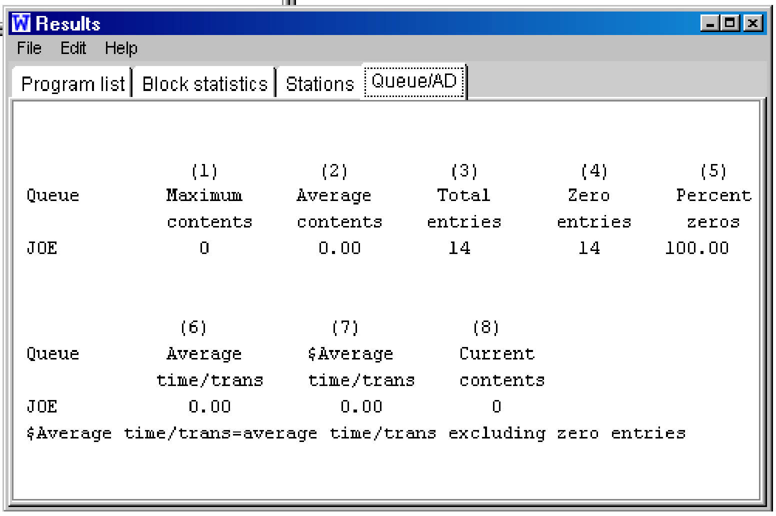
Results – Stations and Queue Statistics

WebGPSS



```
                    (1)           (2)           (3)
Facility          Average      Number of      Average
                utilization     entries     time/trans
JOE               69.21           14          23.73
```

Question #1 Why is it that the barber is busy only about 70% of the time?



```
Queue           (1)         (2)         (3)         (4)         (5)
              Maximum     Average      Total       Zero       Percent
              contents    contents    entries     entries      zeros
JOE              0          0.00         14          14        100.00


Queue          (6)          (7)          (8)
             Average     $Average      Current
            time/trans   time/trans    contents
JOE            0.00         0.00          0
$Average time/trans=average time/trans excluding zero entries
```

Question #2 Why do the queue statistics show that we do not have any queues?

Lesson 17

Simulation Practical Lab Work Activities using WebGPSS

Prepared by : Arya Wirabhuana

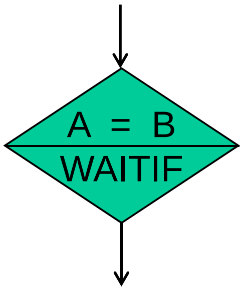
UIN Sunan Kalijaga Yogyakarta

The Lesson 17 Simulation Model

- We will again be making some modifications to our barbershop model.
- We shall now close down the barbershop in a proper manner:
  - At closing time, no new customers will be allowed into the barbershop.
  - Joe will finish cutting the hair of the customer who is getting his/her haircut at closing time.
  - Joe will also give haircuts to the customers who were waiting in the shop at closing time.
- As in our previous barbershop model:
  - Customers arrive randomly, between 12 and 24 minutes apart, and the time for a haircut varies between 20 and 30 minutes, with uniform distributions on these intervals.
  - The barbershop closes its door to new customers after eight hours of operation.

The WAITIF Block

- We need to study a new block , the WAITIF block, in order to achieve the goals of this new model.
- The A operand is the name of a station, either a facility or a storage.
- The B operand is a code dependent on whether the A operand is a facility or a storage, as was the case with the server mode IF block studied in the previous lesson.



If A is a facility, then B can be one of:

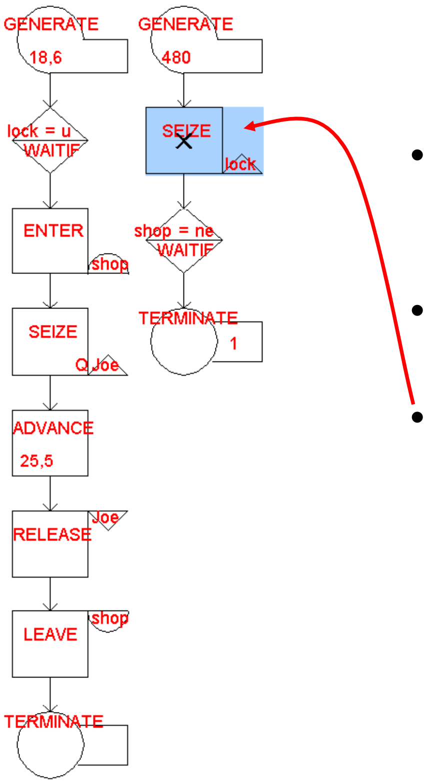
U	Facility in **U**se
NU	Facility **N**ot in **U**se

If A is a storage, then B can be one of:

E	Storage **E**mpty
NE	Storage **N**ot **E**mpty
F	Storage **F**ull
NF	Storage **N**ot **F**ull

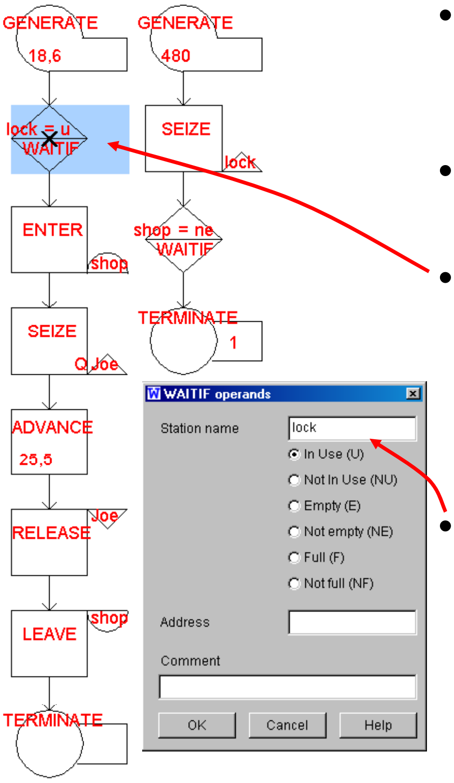
- If, and as long as, the waiting condition, A=B, is true, the transaction has to wait in the block preceding the WAITIF.
- As soon as the condition is not fulfilled, the transaction goes to the block immediately following the WAITIF.
- Checking the wait condition is done when the transaction first comes to the WAITIF and is repeated every time there is an event affecting the station referred to in the WAITIF block.

Step 1: Lock the Door



- We need to have a lock on the door activated at time 480 minutes.
- We accomplish this by an addition to the stop segment.
- This can be done by a “janitor” who arrives at time 480 minutes and “seizes” the lock.

Step 2: Stop New Customers at Closing



Station name: lock

● In Use (U)
○ Not In Use (NU)
○ Empty (E)
○ Not empty (NE)
○ Full (F)
○ Not full (NF)

Address

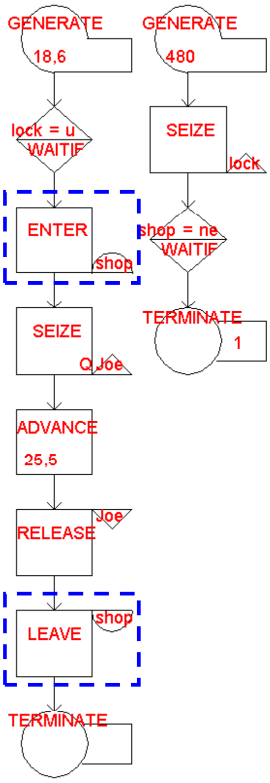
Comment

- We want to prevent new customers from entering once the lock on the door has been activated.
- We accomplish this by an addition to the customer segment.
- A WAITIF block is inserted following the GENERATE block, forcing new customers arriving after the lock has been activated to wait outside of the barbershop.
- The WAITIF operands dialog requires specifying the Station name and the condition to apply.

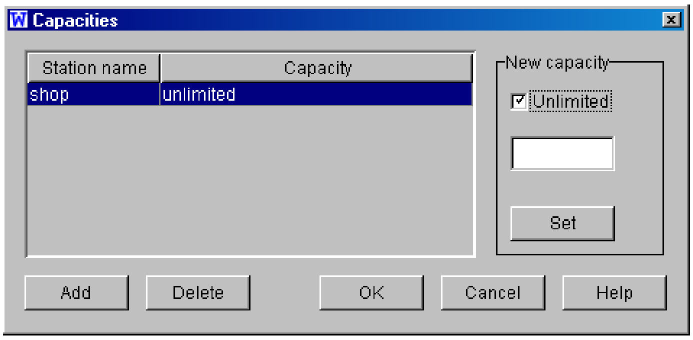
A operand
B operand

WAITIF LOCK=U

Step 3: Keep Track of Customers in Shop



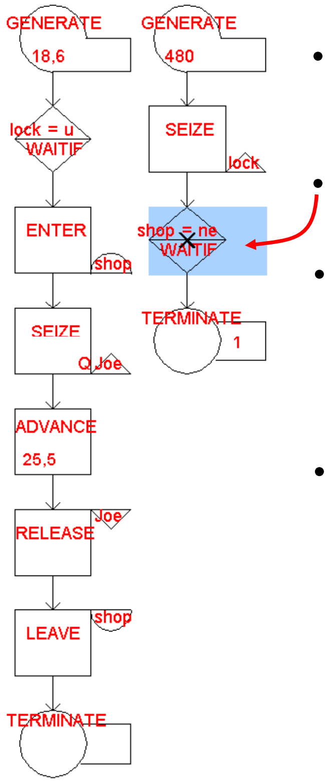
- An ENTER-LEAVE pair of blocks allows us to keep track of all customers that enter and all that leave the barbershop.
- This will allow keeping track of when the storage SHOP becomes empty, i.e., when customers in the shop at closing have completed their haircuts and left the barbershop.
- We must also define the capacity of the storage SHOP, using Control|Capacities from the top menu. We check *Unlimited*, as there is no reason to limit the number of customers inside SHOP at the same time.



Station name	Capacity
shop	unlimited

☑ Unlimited

Step 4: Don’t Stop Till Shop is Empty

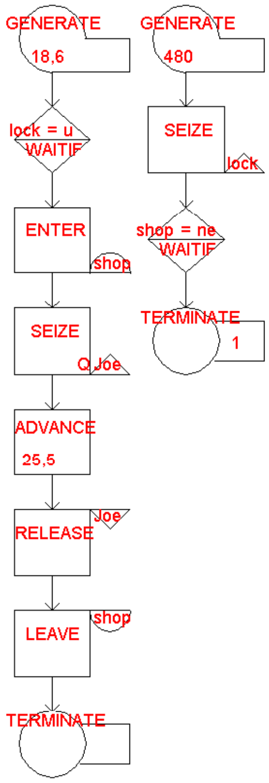


- We now modify the stop segment to keep the simulation from stopping as long as the barbershop is not empty.
- A WAITIF block is inserted prior to the TERMINATE block.
- The janitor waits if, and as long as, the barbershop is **not** empty.

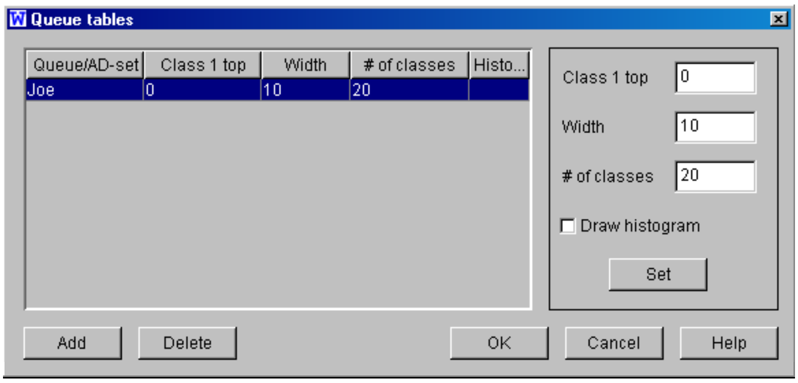
WAITIF SHOP=NE

- As soon as the shop is empty, i.e. when the last customer goes through the LEAVE SHOP block, the janitor proceeds into the TERMINATE block of the stop segment to remove the only token and stop the simulation.

Setting Up a Queue Table for Joe



We also want a queue table for the facility Joe, which we get by clicking on Control|Queue table from the top menu:



Queue/AD-set	Class 1 top	Width	# of classes	Histo...
Joe	0	10	20	

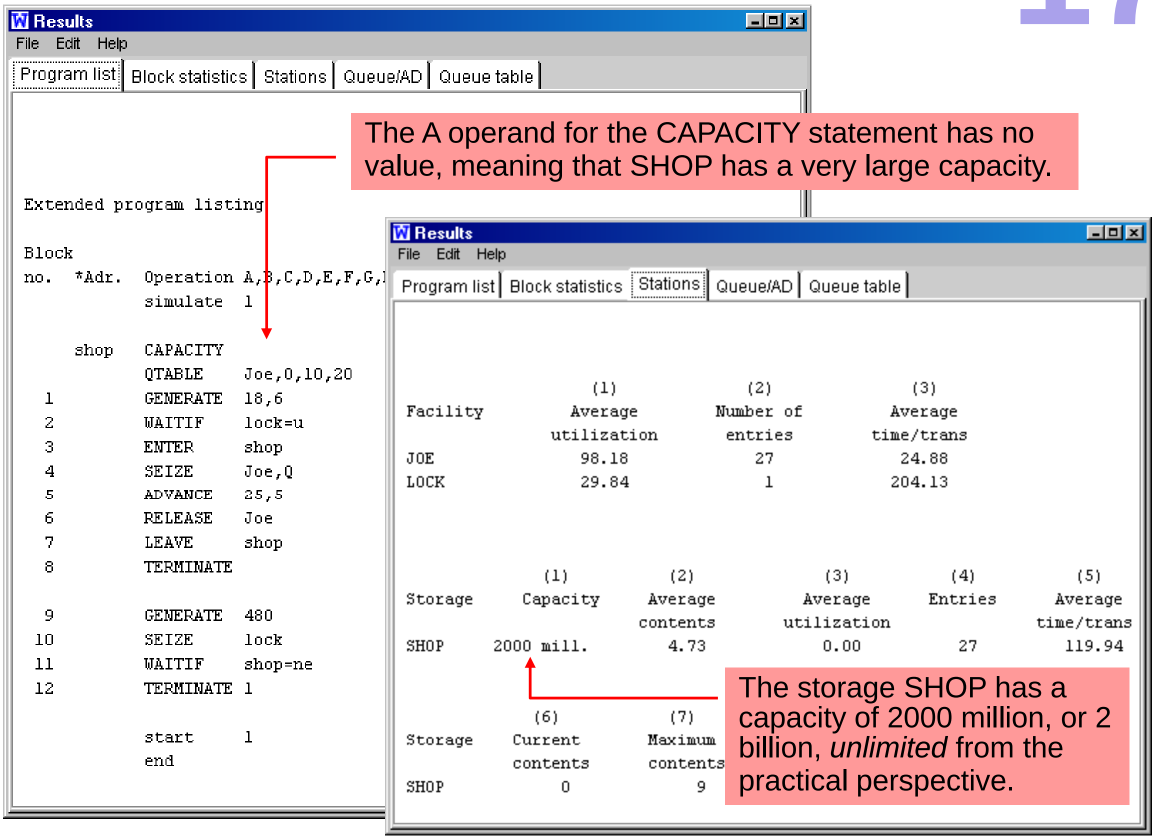
Class 1 top 0

Width 10

# of classes 20

□ Draw histogram

Comments on Default Storage Capacity



```
Extended program listing

Block
no.  *Adr.   Operation A,B,C,D,E,F,G,
               simulate  1

     shop    CAPACITY
             QTABLE    Joe,0,10,20
  1          GENERATE  18,6
  2          WAITIF    lock=u
  3          ENTER     shop
  4          SEIZE     Joe,Q
  5          ADVANCE   25,5
  6          RELEASE   Joe
  7          LEAVE     shop
  8          TERMINATE

  9          GENERATE  480
 10          SEIZE     lock
 11          WAITIF    shop=ne
 12          TERMINATE 1

             start     1
             end
```

The A operand for the CAPACITY statement has no value, meaning that SHOP has a very large capacity.

```
                 (1)            (2)           (3)
Facility       Average       Number of      Average
             utilization      entries      time/trans
JOE             98.18           27           24.88
LOCK            29.84            1          204.13


               (1)         (2)          (3)          (4)        (5)
Storage      Capacity    Average      Average      Entries    Average
                         contents   utilization              time/trans
SHOP        2000 mill.     4.73         0.00          27       119.94


                (6)         (7)
Storage       Current     Maximum
              contents    contents
SHOP             0           9
```

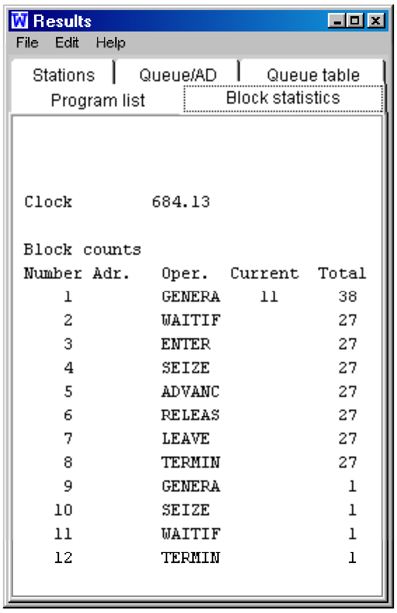
The storage SHOP has a capacity of 2000 million, or 2 billion, *unlimited* from the practical perspective.

Results – Block and Stations Statistics

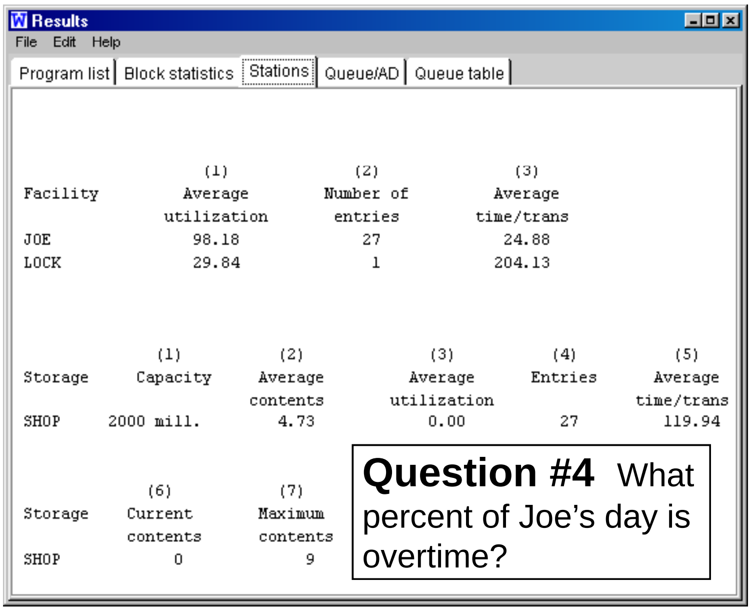
Question #1 How many customers arrived after the door was locked?

Question #2 How many customers were in the shop at simulation stop?

Question #3 How long after closing did Joe remain in the shop? (You can determine this either directly or indirectly depending upon your approach.)



```
Clock        684.13

Block counts
Number Adr.    Oper.   Current   Total
   1           GENERA    11        38
   2           WAITIF              27
   3           ENTER               27
   4           SEIZE               27
   5           ADVANC              27
   6           RELEAS              27
   7           LEAVE               27
   8           TERMIN              27
   9           GENERA               1
  10           SEIZE                1
  11           WAITIF               1
  12           TERMIN               1
```

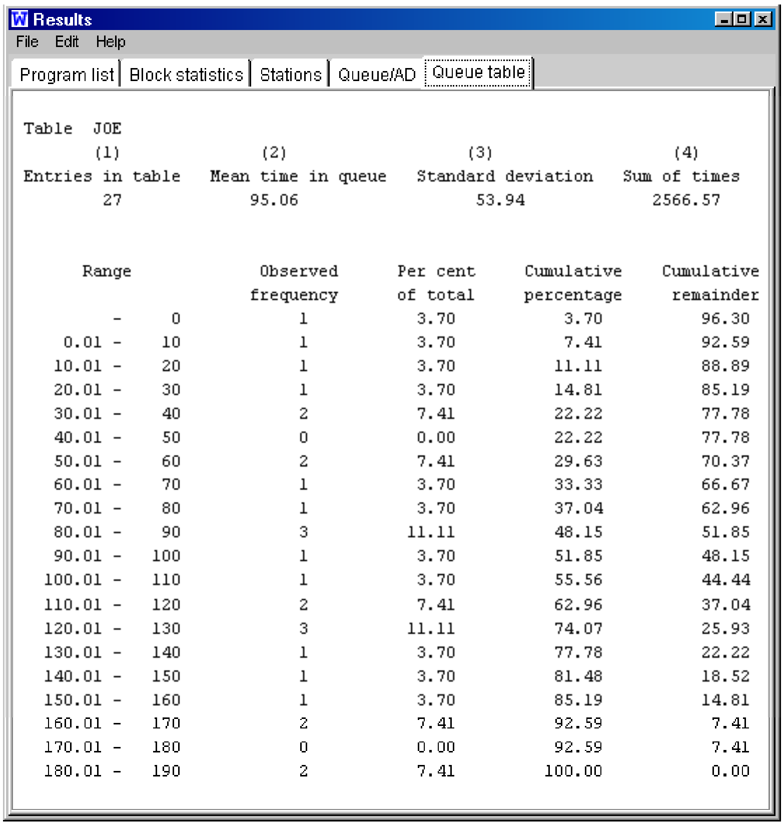


```
                 (1)            (2)           (3)
Facility       Average       Number of      Average
             utilization      entries      time/trans
JOE             98.18           27           24.88
LOCK            29.84            1          204.13


               (1)         (2)          (3)          (4)        (5)
Storage      Capacity    Average      Average      Entries    Average
                         contents   utilization              time/trans
SHOP        2000 mill.     4.73         0.00          27       119.94


                (6)         (7)
Storage       Current     Maximum
              contents    contents
SHOP             0           9
```

Question #4 What percent of Joe's day is overtime?

Results – Queue/AD and Queue Table

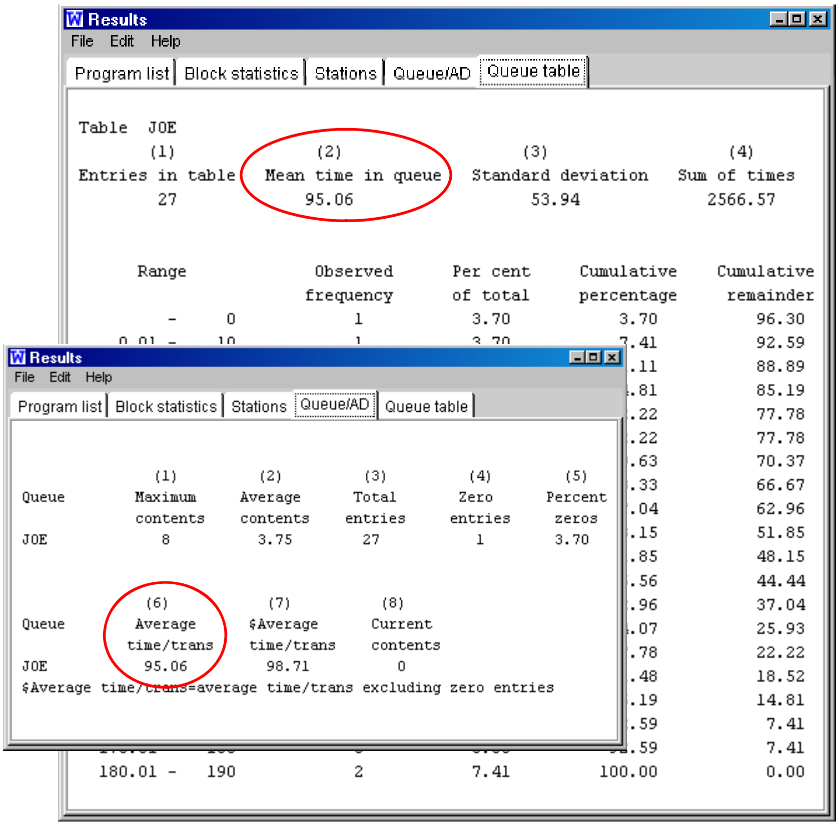
Question
How many customers waited more than 2 hours?



```
Table  JOE
       (1)               (2)                 (3)                 (4)
Entries in table   Mean time in queue   Standard deviation   Sum of times
       27                95.06               53.94              2566.57


       Range            Observed     Per cent     Cumulative    Cumulative
                        frequency    of total     percentage    remainder
          -      0          1          3.70          3.70          96.30
     0.01 -     10          1          3.70          7.41          92.59
    10.01 -     20          1          3.70         11.11          88.89
    20.01 -     30          1          3.70         14.81          85.19
    30.01 -     40          2          7.41         22.22          77.78
    40.01 -     50          0          0.00         22.22          77.78
    50.01 -     60          2          7.41         29.63          70.37
    60.01 -     70          1          3.70         33.33          66.67
    70.01 -     80          1          3.70         37.04          62.96
    80.01 -     90          3         11.11         48.15          51.85
    90.01 -    100          1          3.70         51.85          48.15
   100.01 -    110          1          3.70         55.56          44.44
   110.01 -    120          2          7.41         62.96          37.04
   120.01 -    130          3         11.11         74.07          25.93
   130.01 -    140          1          3.70         77.78          22.22
   140.01 -    150          1          3.70         81.48          18.52
   150.01 -    160          1          3.70         85.19          14.81
   160.01 -    170          2          7.41         92.59           7.41
   170.01 -    180          0          0.00         92.59           7.41
   180.01 -    190          2          7.41        100.00           0.00
```

Results – Queue/AD and Queue Table

Question
Why is the *Mean time in queue* from the queue table the same as the *Average time/trans* from the Queue/AD statistics?



```
Table  JOE
       (1)               (2)                 (3)                 (4)
Entries in table   Mean time in queue   Standard deviation   Sum of times
       27                95.06               53.94              2566.57


       Range            Observed     Per cent     Cumulative    Cumulative
                        frequency    of total     percentage    remainder
          -      0          1          3.70          3.70          96.30
     0.01 -     10          1          3.70          7.41          92.59
                                                      .11          88.89
                                                      .81          85.19
                                                      .22          77.78
                                                      .22          77.78
                                                      .63          70.37
                                                      .33          66.67
                                                      .04          62.96
                                                      .15          51.85
                                                      .85          48.15
                                                      .56          44.44
                                                      .96          37.04
                                                      .07          25.93
                                                      .78          22.22
                                                      .48          18.52
                                                      .19          14.81
                                                      .59           7.41
                                                      .59           7.41
   180.01 -    190          2          7.41        100.00           0.00
```

```
            (1)        (2)        (3)       (4)       (5)
Queue     Maximum    Average     Total     Zero     Percent
          contents   contents   entries   entries    zeros
JOE          8         3.75       27         1        3.70


            (6)        (7)        (8)
Queue     Average    $Average   Current
         time/trans time/trans  contents
JOE        95.06      98.71        0
$Average time/trans=average time/trans excluding zero entries
```

CONTOH KASUS SIMULASI MENGGUNAKAN ARENA

Contoh Kasus –I. Sistem Manufaktur Sederhana.

1. Studi Kasus

Pada suatu sistem perakitan dan tes produk elektronik terdapat dua tipe part yang akan dirakit dengan suatu komponen elektronik. Unit pertama, Part A yang diproduksi pada proses sebelumnya mempunyai waktu antar kedatangan yang dimodelkan dengan distribusi eksponensial dengan mean 5 menit. Selama kedatangan, part tersebut ditransfer pada Part A Prep area, dengan waktu transit 2 menit. Pada Part A Prep area dilakukan proses pembersihan untuk mempersiapkan perakitan dengan waktu proses yang mengikuti distribusi triangular (1, 4, 8). Kemudian part ditransfer ke sealer, dengan waktu transit 2 menit.

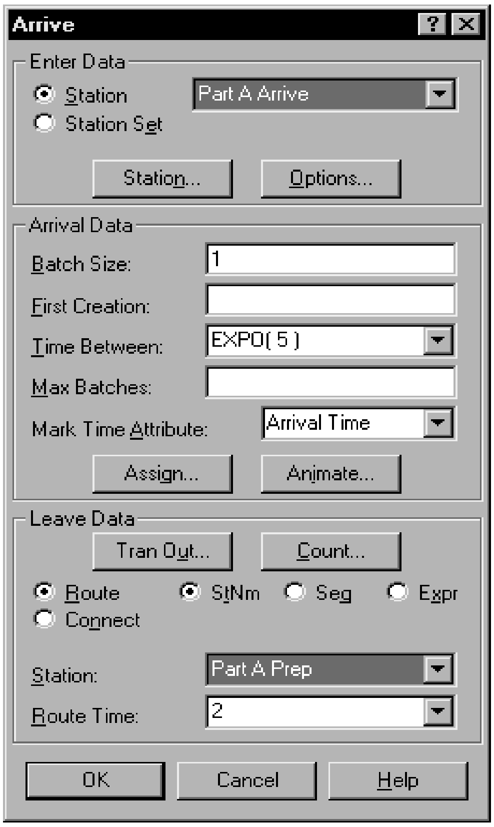
Unit kedua, part B diproduksi pada proses yang berlainan dikelompokkan dalam 4 unit yang kemudian ke sealer. Untuk mendapatkan mengelompokkan Part B dalam jumlah 4 unit dalam tiap batch dibutuhkan waktu yang berdistribusi eksponensial dengan mean 30 menit. Selanjutnya batch dari Part B diproses pada Part B Prep dengan waktu proses distribusi triangular (3, 5, 10). Kemudian batch tersebut dikirim ke sealer dengan waktu transit 2 menit.

Pada operasi sealer, perakitan komponen tergantung pada tipe part yang diproses. Part A membutuhkan waktu proses dengan distribusi triangular (1,3,4) dan Part B membutuhkan waktu pelayanan yang berdistribusi normal dengan rata-rata 2,4 menit dan standar deviasi 0.5 menit. 91% part yang diperiksa dinyatakan baik dan langsung dikirim pada shipping departemen. Part yang tidak lolos dari proses pemeriksaan ditransfer pada rework area dimana diadakan dissambled, repaired, clened, assembled, dan re-tested. 80 % part yang reject berhasil diperbaiki dan ditransfer pada shipping departemen. Part yang tidak dapat diperbaiki ditransfer pada scrap area. Waktu untuk me-rework part mengikuti distribusi eksponensial dengan mean 45 menit dan tidak tergantung pada tipe part atau status part, dan kerusakannya. Diasumsikan semua waktu transfer membutuhkan 2 menit.

2. Pembangunan Model

Karena terdapat dua tipe yang berbeda pada kedatangan entiti dengan pola yang berbeda pada mesing-mesing tipe, disini akan digunakan dua Arrive module yang berbeda (masing-masing untuk tiap tipe) untuk membangkitkan kedatangan part. Tiap Prep area dimodelkan dalam Server module. Operasi sealer termasuk pemeriksaan terhadap parts yang menghasilkan dua kemungkinan good dan reject. Sehingga digunakkan Inspect module yang sesuai dengan kapabilitasnya, dimana berdasarkan pada Server module dengan output pass dan fail. Rework area juga dimodelkan dalam Inspect module, yang juga mempunyai pilihan outputpass dan fail. Part yang meninggalkan sistem dimodelkan dalam tiga Depart module yang terpisah (shipped, salvage, dan scrapped) sehingga dapat dihitung masing-masing flowtime.

Untuk memodelkan transit antar stasiun, digunakan pilihan Route untuk menghubungkan module Arrive, Server, dan Inspect dengan waktu route 2 menit pada masing-masing transit. Ditambahkan Simulate module untuk mendefenisikan panjang simulasi dan gambar animasi part. Semua module yang digunakan didapat pada common



● Station ○ Station Set: Part A Arrive

Batch Size: 1

First Creation:

Time Between: EXPO(5)

Max Batches:

Mark Time Attribute: Arrival Time

● Route ○ Connect

● StNm ○ Seq ○ Expr

Station: Part A Prep

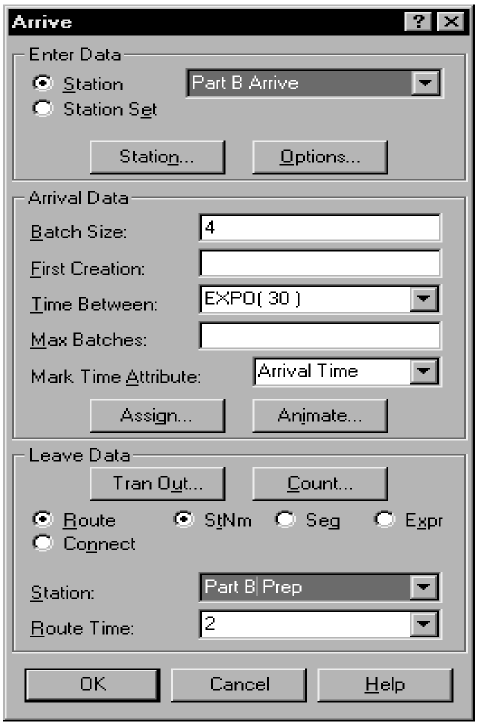
Route Time: 2

panel. Route yang digunakandari Animate toolbar untuk menunjukkan pergerakan part dalam sistem.

Pada proses pembuatan model pada tiap modul yang digunakan dimasukkan informasi yang dibutuhkan oleh setiap modul. Pertama pada kedatangan part yang dimodelkan dalam Arrive modul.

Sebelum keluar dari dialog module Arrive didefenisikan terlebih dahulu attribute Sealer Time dan di-assign nilai dari distribusi triangular (1,3,4). Untuk mengerjakan ini, klik pada button Assign, kemudian pada Add pada kotak dialog baru dan masukkan informasi yang dibutuhkan.

Arrive module untuk kedatangan Part B hampir sama dengan pada Part A kecuali beberapa informasi yang dibutuhkan berbeda seperti pada dibawah ini :



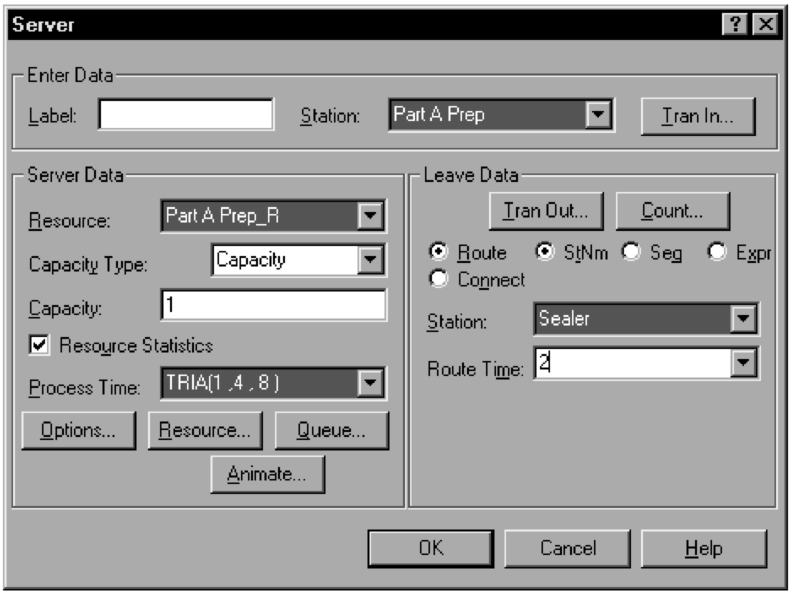
● Station: Part B Arrive
○ Station Set

Batch Size: 4
First Creation:
Time Between: EXPO(30)
Max Batches:
Mark Time Attribute: Arrival Time

● Route ● StNm ○ Seq ○ Expr
○ Connect

Station: Part B Prep
Route Time: 2

Setelah data untuk kedatangan part maka langkah selanjutnya adalah entry untuk preparation area. Untuk preparation area dibangun oleh module server dengan entry data sebagai berikut :


Server

Enter Data

Label:

Station: Part A Prep

Server Data

Resource: Part A Prep_R

Capacity Type: Capacity

Capacity: 1

☑ Resource Statistics

Process Time: TRIA(1 ,4 , 8)

Leave Data

● Route ● StNm ○ Seg ○ Expr

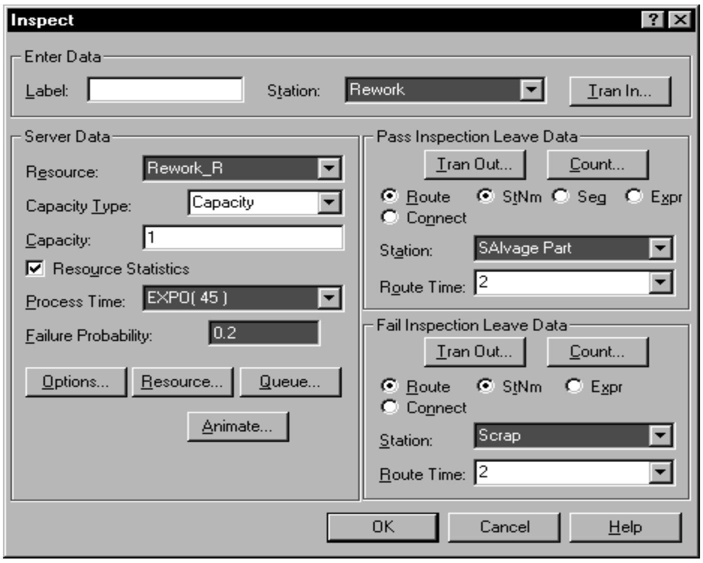
○ Connect

Station: Sealer

Route Time: 2

Langkah selanjutnya adalah entry data untuk operasi sealer, dimana untuk operasi ini digunakan Inspect module. Inspect module mempunyai dua jalan untuk entity yang meninggalkan mudule Pass atau Fail. Data yang dimasukkan adalah :

Part yang melalui stasiun inspeksi pertama langsung menuju stasiun shipping, sedangkan yang reject menuju stasiun Rework. Sehingga dibutuhkan tambahan stasiun Rework yang dibuat dari Inspect module.


Inspect

Enter Data

Label:

Station: Rework

Server Data

Resource: Rework_R

Capacity Type: Capacity

Capacity: 1

☑ Resource Statistics

Process Time: EXPO(45)

Failure Probability: 0.2

Pass Inspection Leave Data

● Route ● StNm ○ Seg ○ Expr

○ Connect

Station: SAlvage Part

Route Time: 2

Fail Inspection Leave Data

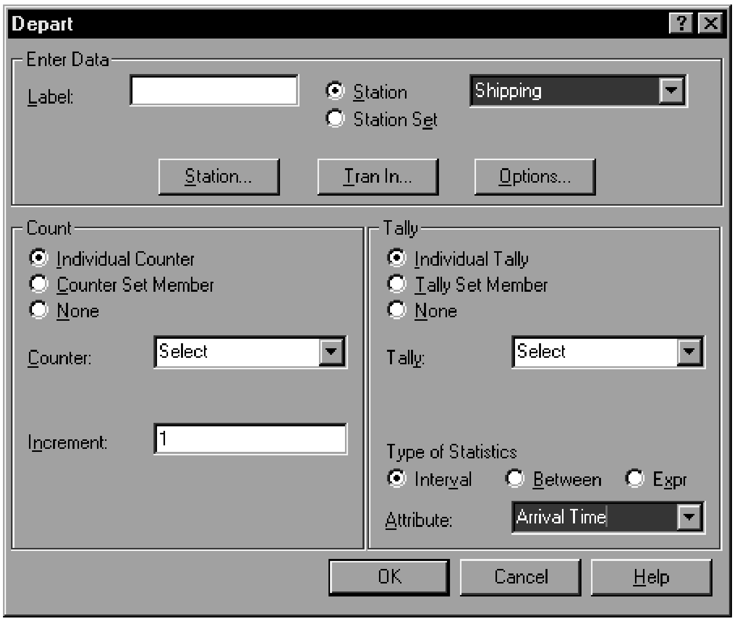
● Route ● StNm ○ Expr

○ Connect

Station: Scrap

Route Time: 2

Setelah part melalui semua operasi tersebut, maka part meniggalkan sistem dimana pada model ini dibangun dengan Depart module. Terdapat 3 stasiun dimana part meninggalkan operasi berdasarkan tempat operasinya yaitu Shipping, Salveged Part, dan Scrap. Data untuk mesing-masing stasiun adalah sebagai berikut :



Depart

Enter Data

Label:

● Station Shipping

○ Station Set

Count

● Individual Counter

○ Counter Set Member

○ None

Counter: Select

Increment: 1

Tally

● Individual Tally

○ Tally Set Member

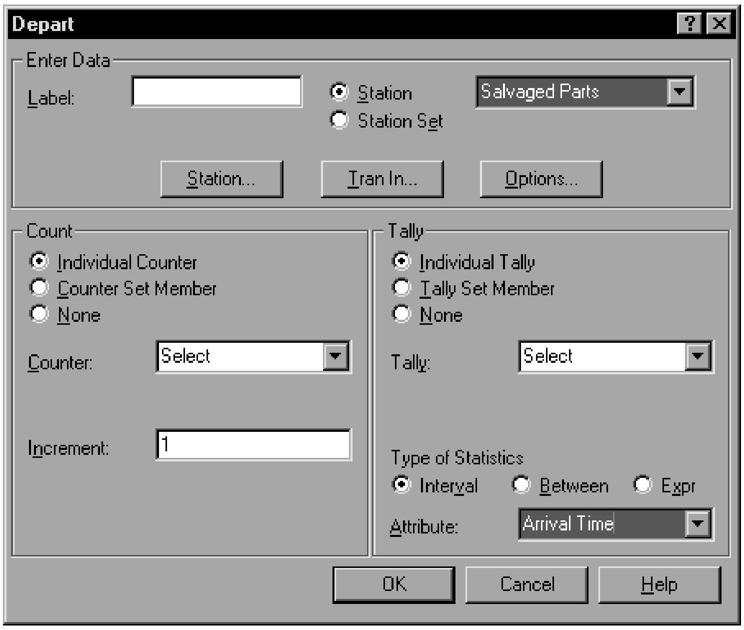
○ None

Tally: Select

Type of Statistics

● Interval ○ Between ○ Expr

Attribute: Arrival Time



Depart

Enter Data

Label:

● Station Salvaged Parts

○ Station Set

Count

● Individual Counter

○ Counter Set Member

○ None

Counter: Select

Increment: 1

Tally

● Individual Tally

○ Tally Set Member

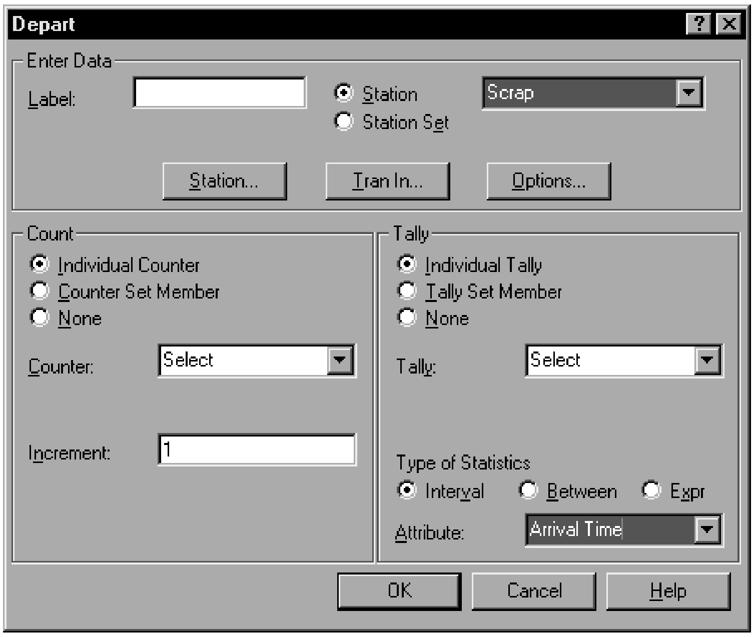
○ None

Tally: Select

Type of Statistics

● Interval ○ Between ○ Expr

Attribute: Arrival Time



Depart

Enter Data

Label:

● Station
○ Station Set

Scrap

Count

● Individual Counter
○ Counter Set Member
○ None

Counter: Select

Increment: 1

Tally

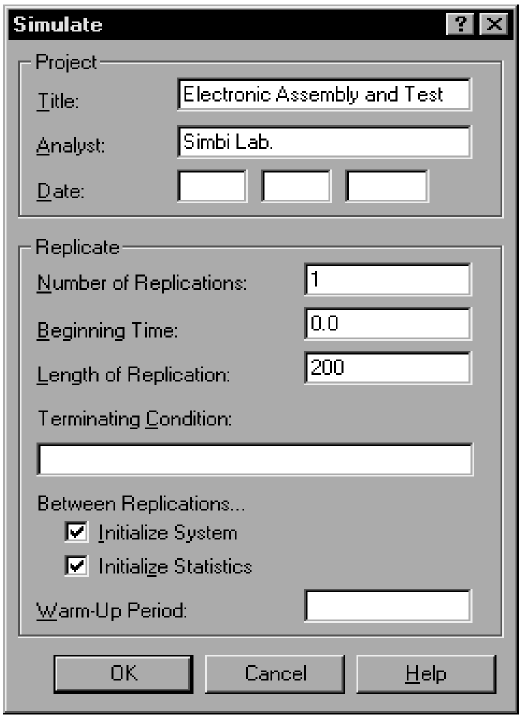
● Individual Tally
○ Tally Set Member
○ None

Tally: Select

Type of Statistics
● Interval ○ Between ○ Expr

Attribute: Arrival Time

Module terakhir yang digunakan setelah semua operasi digambarkan adalah Simulate module. Pada module ini data yang dimasukkan hanya meliputi panjang waktu simulasi dan informasi mengenai project.



Simulate

Project

Title: Electronic Assembly and Test

Analyst: Simbi Lab.

Date:

Replicate

Number of Replications: 1

Beginning Time: 0.0

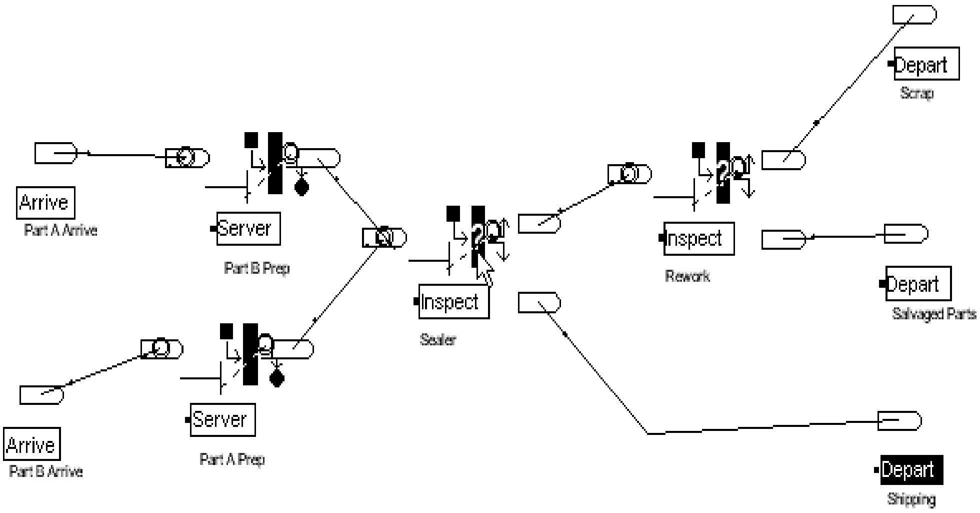
Length of Replication: 200

Terminating Condition:

Between Replications...
☑ Initialize System
☑ Initialize Statistics

Warm-Up Period:

Langkah terakhir dalam membangun model adalah menambahkan route path yang menggambarkan jalannya pergerakkan entity selama simulasi berlangsung. Route path dipilih dari Animate toolbar dan digunakan untuk menghubungkan segmen-segmen.



3. Report Summary

Setelah tahapan pembangunan selesai dikerjakan selanjutnya dapat diadakan pemeriksaan terhadap model yang dibuat untuk melihat apakah terdapat kesalahan pada model yang dibuat. Apabila tidak terdapat kesalahan maka program simulasi dapat dijalankan dan melihat report summary dari hasil simulasi tersebut. Berikut adalah hasil / report summary dari pemodelan yang telah kita lakukan.

```
Summary for Replication 1 of 1
Project:  Electronic Assem                     Run execution date :   8/ 9/1999
Analyst:  SIMBI LAb.                           Model revision date:   8/ 9/1999

Replication ended at time      : 200.0

                                 TALLY VARIABLES

Identifier              Average   Half Width  Minimum    Maximum   Observations
_______________________________________________________________________________

Salvaged Parts_Ta       107.87     (Insuf)    101.91     113.82          2
Rework_R_Q Queue Time   30.164     (Insuf)    .00000     40.174          5
Sealer_R_Q Queue Time   1.6574     (Insuf)    .00000     7.0454         59
Scrap_Ta                86.079     (Insuf)    76.343     95.814          2
Part B Prep_R_Q Queue   31.124     (Insuf)    .00000     67.188         30
Shipping_Ta             30.858     (Insuf)    10.988     85.964         48
Part A Prep_R_Q Queue   3.0383     (Insuf)    .00000     8.6960         31

                           DISCRETE-CHANGE VARIABLES

Identifier              Average   Half Width  Minimum    Maximum   Final Value
_______________________________________________________________________________
```

```
# in Rework_R_Q        2.4126     (Insuf)    .00000     5.0000     5.0000
Part A Prep_R Busy      .66022     (Insuf)    .00000     1.0000     1.0000
# in Sealer_R_Q         .48893     (Insuf)    .00000     4.0000     .00000
Part B Prep_R Availabl  1.0000     (Insuf)    1.0000     1.0000     1.0000
# in Part A Prep_R_Q    .47094     (Insuf)    .00000     3.0000     .00000
Sealer_R Available      1.0000     (Insuf)    1.0000     1.0000     1.0000
Part A Prep_R Availabl  1.0000     (Insuf)    1.0000     1.0000     1.0000
Rework_R Available      1.0000     (Insuf)    1.0000     1.0000     1.0000
Sealer_R Busy           .75942     (Insuf)    .00000     1.0000     .00000
Rework_R Busy           .90293     (Insuf)    .00000     1.0000     1.0000
Part B Prep_R Busy      .93654     (Insuf)    .00000     1.0000     1.0000
# in Part B Prep_R_Q    5.7990     (Insuf)    .00000     11.000     10.000

                                   COUNTERS

                    Identifier                Count   Limit
                    _________________________________________

                    Salvaged Parts_C              2  Infinite
                    Scrap_C                       2  Infinite
                    Shipping_C                   48  Infinite


Simulation run time: 0.10 minutes.
Simulation run complete.
```

4. Pemodelan Detail

Untuk dapat melihat sistem proses lebih detail maka waktu simulasi akan digunakan berdasarkan panjang waktu satu shift kerja. Sister secara actual beroperasi sebanyak dua shift perhari. Dari pengamatan awal ternyata operasi pada shift pertama dan menambah operator pada shift kedua menjadi dua orang.

Ternyata pada sealer operation terdapat permasalahan pada mesinnya atau mengalami failure sehingga pada operasi tersebut mengalami bottleneck. Mesin rusak secara periodik. Setelah dikumpulkan data maka kerusakan mesin mempunyai mean uptime beetwen failure 120 menit dengan distribusi eksponensial. Perbaikan mesin tersebut membutuhkan waktu yang berdistribusi eksponensial dengan mean 4 menit.

Langkah selanjutnya adalah memodifikasi model dengan memasukkan dua aspek tersebut.

5. Resource Schedule

Untuk menjadwalkan resource pada operasi rework langkah-langkah modifikasi adalah sebagai beikut :

- Aktifkan kotak dialog Rework operation dan masukkan data berikut :

```
Server Data
      Capacity Type                        Schedule
      Schedule                       Rework Schedule
```

- Pilih button Schedule, tambahkan data berikut :

Capacity	1
Duration	480
Capacity	2
Duration	480

6.Resource Failure

Untuk memodelkan failure pada mesin, klik pada Failure Add button dan buka dialog Failure seperti pada gambar. Kemudian masukkan data sebagai berikut :

Failure	Sealer Failure
Failure Based On	Time
Fail When	Wait
Uptime	EXPO (120)
Downtime	EXPO (4)

7. Perubahan Animasi

Animasi yang ditampilkan katika model dijalankan merupakan default yang disediakan oleh Arena. Untuk lebih interaktif maka animasi dari tiap module dapat dirubah melalui penambahan gambar dengan fasilitas draw atau mewngedit gambar default yang ditampilkan.

1. Merubah Animasi Antrian

   Animasi antrian dapat dirubah melalui perubahan ukuran lintasan antrian banyaknya point yang dimuat dalam lintasan tersebut. Untuk merubah panjang antyrian dilakukan dengan memperpanjang garis antrian. Double-click pada garis tersebut akan menampilkan kotak dialog antrian. Selanjutnya pada option type pilih point. Klik button point, pada kotak dialog point klik Add untuk menambahkan jumlah point.

2. Merubah Gambar Entity

   Perubahan entity picture harus didefenisikan pada module yang melibatkan entity. Untuk model yang dibuat terdapat dua jenis entity yaitu Part A dan Part B. Stasiun yang terlibat yaitu Arrive, Part Prep, Sealer, Rework.

3. Animasi Stasiun Arrive

Buka kotak dialog module Arrive. Untuk mendefenisikan nama gambar yang dipakai klik pada bitton Animate. Setelah muncuil kotak dialog Entity Anomation Options masukkan data :

Initial Entity	
Picture	Select
Initial Picture	Part A

Demikian juga untuk part B

Initial Entity	
Picture	Select
Initial Picture	Bacth B

4. Perubahan Animasi untuk Stasiun Part B Prep

Buka kotak dialog Server module klik pada button Animate. Setelah muncul kotak dialog Entity Animation Option, masukkan data :

Change Picture When Enter	
Picture	Part B

Ketika mendefenisikan entity picture harus bewrhubungan dengan gambar yang telah diberi suatu nama sesuai dengan defenisi pada modul. Pembuatan gambar entity dilakukan dengan double-click pada gambar entity pada module Simulate. Selanjutnya akan ditampilkan entity Picture Plecement.

Untuk menambah gambar entity klik Add kemudian copy. Double-click pada gambar akan menampilkan editing gambar entity.

Ubah gambar entity sebagai berikut. Nilai value ditetapkan dari kotak pull-down.

5. Merubah Animasi Resources

Buka jendela Resource Picture Elemen Placement (dengan Double-klik pada resource Picture) untuk mengganti gambar resource buka jendela edit gambar (dengan double-klik pada gambar). Dengan fasilitas drawing ubah gambar resource . Lakukan pada masing-masing stasiun.

8. Report/Summary Final

Setelah dilakukan pemodelan detail dan beberapa perubahan, maka kita jalankan kembali simulasi dengan jumlah replikasi sebanyak kali. Berikut ini adalah hasilnya :

```
Summary for Replication 1 of 2
Project:  Electronic Assem                     Run execution date :   8/ 9/1999
Analyst:  Mr.Munchkin                          Model revision date:   8/ 9/1999
Replication ended at time      : 480.0

                                 TALLY VARIABLES
Identifier              Average   Half Width  Minimum    Maximum   Observations
_____________________________________________________________________________

Salvaged Parts_Ta       163.13     (Insuf)    26.167     259.26           8
Rework_R_Q Queue Time   114.39     (Insuf)    .00000     233.03          11
Sealer_R_Q Queue Time   1.7596     (Insuf)    .00000     8.2380         145
Scrap_Ta                189.48     (Insuf)    189.48     189.48           1
Part B Prep_R_Q Queue   14.184     (Insuf)    .00000     43.220          58
Shipping_Ta             23.762     (Insuf)    11.365     53.456         127
Part A Prep_R_Q Queue   4.9051     (Insuf)    .00000     20.283          89

                           DISCRETE-CHANGE VARIABLES
Identifier              Average   Half Width  Minimum    Maximum   Final Value
_______________________________________________________________________________

# in Rework_R_Q         2.9346     (Insuf)    .00000     7.0000     5.0000
Part A Prep_R Busy      .83422     (Insuf)    .00000     1.0000     1.0000
# in Sealer_R_Q         .53155     (Insuf)    .00000     3.0000     .00000
Part B Prep_R Availabl  1.0000     (Insuf)    1.0000     1.0000     1.0000
# in Part A Prep_R_Q    .90950     (Insuf)    .00000     5.0000     .00000
Sealer_R Available      1.0000     (Insuf)    1.0000     1.0000     1.0000
Part A Prep_R Availabl  1.0000     (Insuf)    1.0000     1.0000     1.0000
Rework_R Available      1.0000     (Insuf)    1.0000     2.0000     2.0000
Sealer_R Busy           .75905     (Insuf)    .00000     1.0000     1.0000
Rework_R Busy           .84868     (Insuf)    .00000     2.0000     2.0000
Part B Prep_R Busy      .74129     (Insuf)    .00000     1.0000     1.0000
# in Part B Prep_R_Q    1.9144     (Insuf)    .00000     9.0000     2.0000

                                   COUNTERS
                    Identifier                Count   Limit
                    _________________________________________

                    Salvaged Parts_C              8  Infinite
                    Scrap_C                       1  Infinite
                    Shipping_C                  127  Infinite


Beginning replication 2 of 2

                              Summary for Replication 2 of 2

Project:  Electronic Assem                     Run execution date :   8/ 9/1999
Analyst:  Mr.Munchkin                          Model revision date:   8/ 9/1999
Replication ended at time      : 480.0

                                 TALLY VARIABLES
Identifier              Average   Half Width  Minimum    Maximum   Observations
_______________________________________________________________________________

Salvaged Parts_Ta       243.26     (Insuf)    146.25     410.49           3
Rework_R_Q Queue Time   117.03     (Insuf)    .00000     253.51           5
Sealer_R_Q Queue Time   29.970     (Insuf)    .00000     40.978         177
Scrap_Ta                    --         --         --         --           0
Part B Prep_R_Q Queue   66.776     (Insuf)    .00000     143.51          80
Shipping_Ta             74.672     (Insuf)    10.745     193.06         166
Part A Prep_R_Q Queue   12.251     (Insuf)    .00000     30.074         110
```

```
                           DISCRETE-CHANGE VARIABLES
Identifier              Average   Half Width  Minimum    Maximum   Final Value
_______________________________________________________________________________

# in Rework_R_Q         2.3879     (Insuf)    .00000     7.0000     5.0000
Part A Prep_R Busy      .93483     (Insuf)    .00000     1.0000     1.0000
# in Sealer_R_Q         11.387     (Corr)     .00000     16.000     11.000
Part B Prep_R Availabl  1.0000     (Insuf)    1.0000     1.0000     1.0000
# in Part A Prep_R_Q    2.9051     (Insuf)    .00000     8.0000     6.0000
Sealer_R Available      1.0000     (Insuf)    1.0000     1.0000     1.0000
Part A Prep_R Availabl  1.0000     (Insuf)    1.0000     1.0000     1.0000
Rework_R Available      1.0000     (Insuf)    1.0000     2.0000     2.0000
Sealer_R Busy           .92052     (Insuf)    .00000     1.0000     1.0000
Rework_R Busy           .79787     (Insuf)    .00000     2.0000     2.0000
Part B Prep_R Busy      .99583     (Insuf)    .00000     1.0000     1.0000
# in Part B Prep_R_Q    13.423     (Insuf)    .00000     26.000     12.000

                                   COUNTERS
                    Identifier             Count   Limit
                    _________________________________________

                    Salvaged Parts_C           3  Infinite
                    Scrap_C                    0  Infinite
                    Shipping_C               166  Infinite

Simulation run time: 0.62 minutes.
Simulation run complete.
```

Contoh Kasus-II. Antrian Multiserver

Kasus ini menerangkan bagaimana entiti dari sistem akan memasuki sebuah proses dengan jumlah stasiun pelayanan lebih dari satu. Dengan demikian entiti akan menentukan stasiun pelayanannya, untuk itu dia harus menentukan lini antrian yang akan dimasuki sesuai dengan preferensi prioritasnya. Kita akan menganalisis bgaimana utilitas setiap server/pelayan, dan bagaimana status jumlah sumber daya yang ada (kurang atau lebih).

Module yang digunakan : Arrive, Enter, PickQueue, Process, Leave, Depart,
Simulate, dan Animate.

<u>*1. Model Verbal.*</u>

Dalam sebuah minimarket terdapat empat orang yang bertugas sebagai kasir. Manager minimarket tersebut ingin mengetahui apakah dengan empat kasir tersebut sudah cukup untuk manangani para konsumen yang berbelanja apabila diasumsikan bahwa komsumen dapat mentolerir panjang antrian maksimum sebanyak 5 orang.

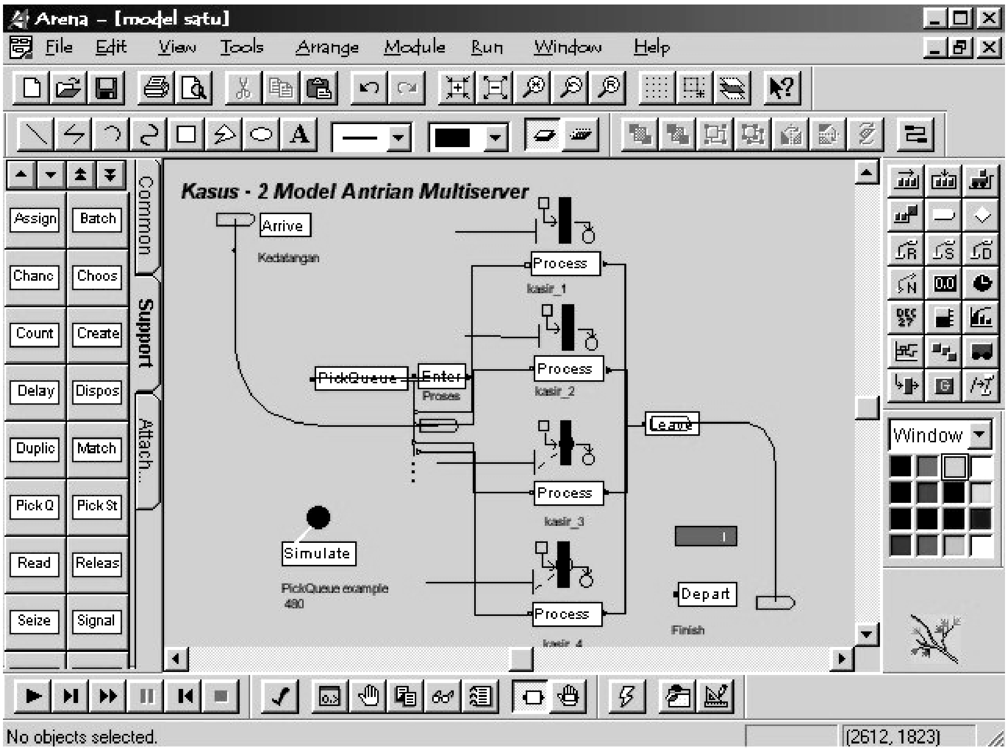
Dari penelitian dapat diambil sebuah kesimpulan para konsumen memasuki kasir dengan waktu rata-rata 1,5 menit dengan distribusi eksponensial. Waktu yang ditempuh oleh para konsumen dari rak-rak produk ke kasir memiliki rata-rata 3 menit. Para konsumen akan dilayani oleh empat orang kasir. Kasir pertama memiliki kemampuan melayani dengan distribusi normal dengan rata-rata 5 menit dan standar deviasi 4 menit. Kasir kedua memiliki waktu pelayanan yang berdistribusi triangular (2,4,5). Kasir ketiga memiliki waktu pelayanan yang berdistribusi eksponensial 6 menit. Sedangkan kasir keempat memiliki kemampuan melayani dengan distribusi poisson 5 menit.

.2. Pembangunan Model ARENA

Karena hanya terdapat satu pola kedatangan entiti pada sistem yang dimodelkan, maka kita gunakan satu buah module arrive untuk membangkitkan kedatangan entiti. Berikutnya ada beberapa fasilitas pelayanan yang digunakan untuk melayani entiti mengikuti model antrian multiserver, untuk itu kita menggunakan modile pickqueue untuk memilih mekanisme antriannya. Berikutnya setiap fasilitas pelayanan diasumsikan sebagi sebuah proses tersendiri, namun secara kolektif merupakan sebuah kesatuan kegiatan yang dilakukan oleh fasilitas pelayanan tersebut. Kita akan menggunakan module enter untuk menandai masuknya entiti pada proses pelayanan dan module process untuk setiap fasilitas pelayanan yang ada. Module leave digunakan untuk tanda bahwa entiti telah meninggalkan fasilitas pelayanan dan model diakhiri dengan module depart untuk pernyataaan bahwa entiti telah meninggalkan sistem yang diamati.

Untuk memodelkan transit antar stasiun, digunakan pilihan Route untuk menghubungkan module Arrive, Enter, Leave, dan Depart. Sedangkan untuk menghubungkan logika prosedurnya kita menggunakan pilihan connect. Ditambahkan Simulate module untuk mendefinisikan panjang simulasi dan gambar animasi part. Untuk module arrive, enter, process, leave, dan depart didapat pada common panel , sedangkan module pickqueue didapat dari support panel. Untuk route dipilih dari animate toolbar untuk menggambarkan pergerakan part dalam sistem. Sedangkan untuk connect dipilih dari arrange toolbar yang berfungsi menghubungkan module yang diikuti oleh entiti.

Berikut ini adalah gambaran model awal (tanpa animasi) dari kasus diatas :

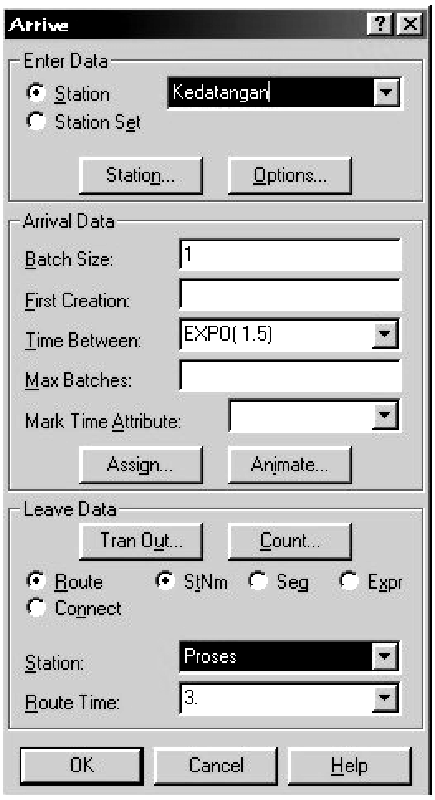


No objects selected.

Setelah Model tersebut selesai, kita dapat mengubah dari bentuk standartnya menjadi bentuk yang mendekati sistem nyata dengan cara mengoptimalkan fungsi animasi dan *drawing* yang dimiliki oleh software ARENA ini.

Berikut adalah Gambar panduan dalam pembuatan model :

1. Module Arrive – Kedatangan



Arrive

Enter Data

● Station ○ Station Set: Kedatangan

Arrival Data

Batch Size: 1

First Creation:

Time Between: EXPO(1.5)

Max Batches:

Mark Time Attribute:

Leave Data

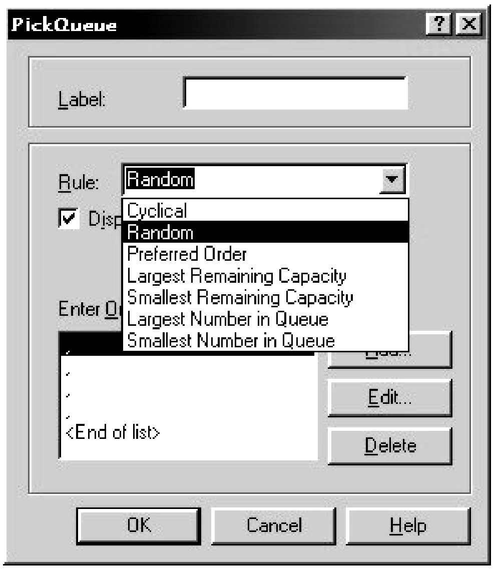
● Route ○ Connect

● StNm ○ Seq ○ Expr

Station: Proses

Route Time: 3.

2. Module Pick Queue – Pemilihan Lini Antrian



PickQueue

Label:

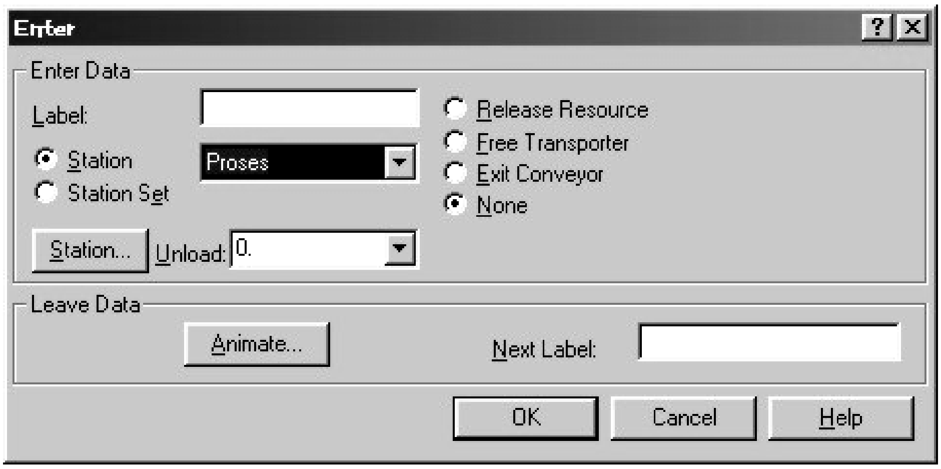
Rule: Random

☑ Disp

Enter Q

<End of list>

3. Module Enter – Masuk ke lini antrian


Enter

Enter Data

Label:

● Station: Proses

○ Station Set

Station... Unload: 0.

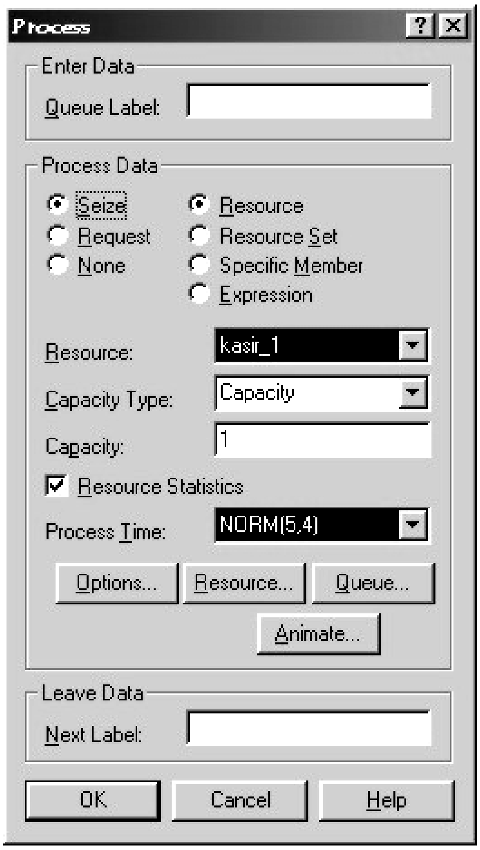
○ Release Resource
○ Free Transporter
○ Exit Conveyor
● None

Leave Data

Animate... Next Label:

OK Cancel Help

4. Module Process – Proses pelayanan


Process

Enter Data

Queue Label:

Process Data

● Seize ○ Request ○ None

● Resource ○ Resource Set ○ Specific Member ○ Expression

Resource: kasir_1

Capacity Type: Capacity

Capacity: 1

☑ Resource Statistics

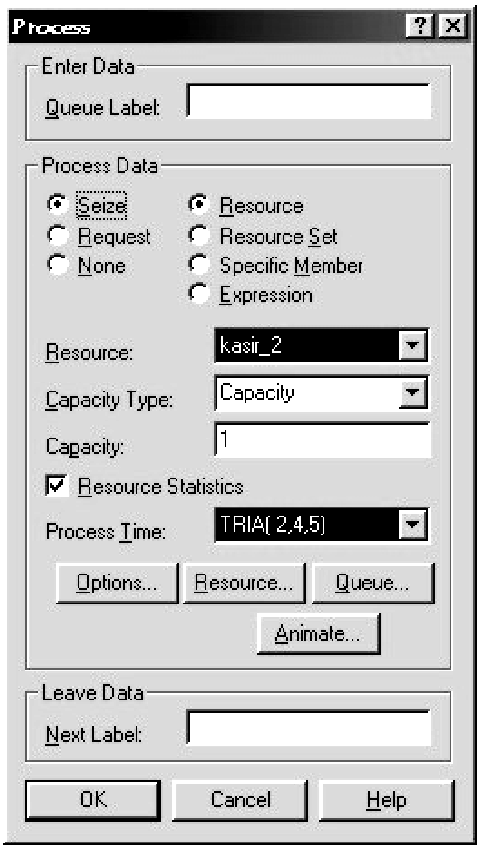
Process Time: NORM(5,4)

Options... Resource... Queue... Animate...

Leave Data

Next Label:

OK Cancel Help


Process

Enter Data

Queue Label:

Process Data

● Seize ○ Request ○ None

● Resource ○ Resource Set ○ Specific Member ○ Expression

Resource: kasir_2

Capacity Type: Capacity

Capacity: 1

☑ Resource Statistics

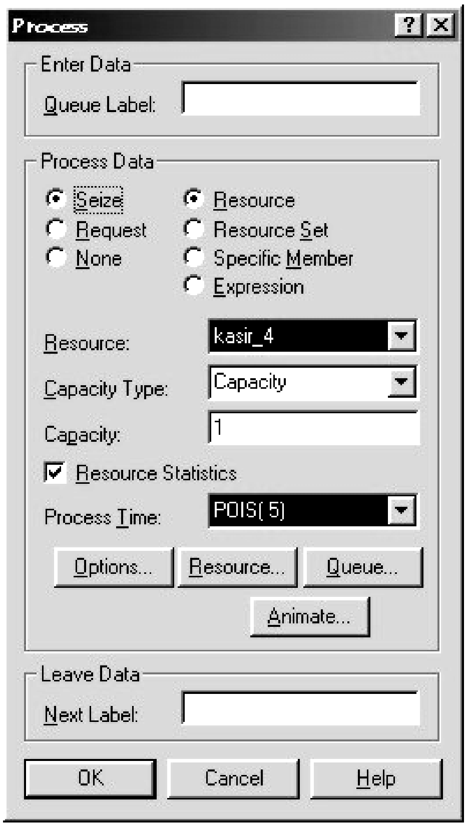
Process Time: TRIA(2,4,5)

Options... Resource... Queue... Animate...

Leave Data

Next Label:

OK Cancel Help



Queue Label:

● Seize ○ Request ○ None

● Resource ○ Resource Set ○ Specific Member ○ Expression

Resource: kasir_4

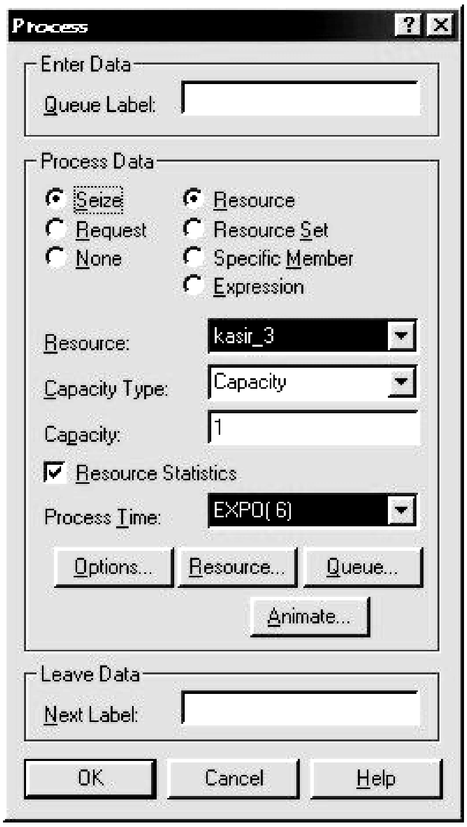
Capacity Type: Capacity

Capacity: 1

☑ Resource Statistics

Process Time: POIS(5)

Next Label:



Queue Label:

● Seize ○ Request ○ None

● Resource ○ Resource Set ○ Specific Member ○ Expression

Resource: kasir_3

Capacity Type: Capacity

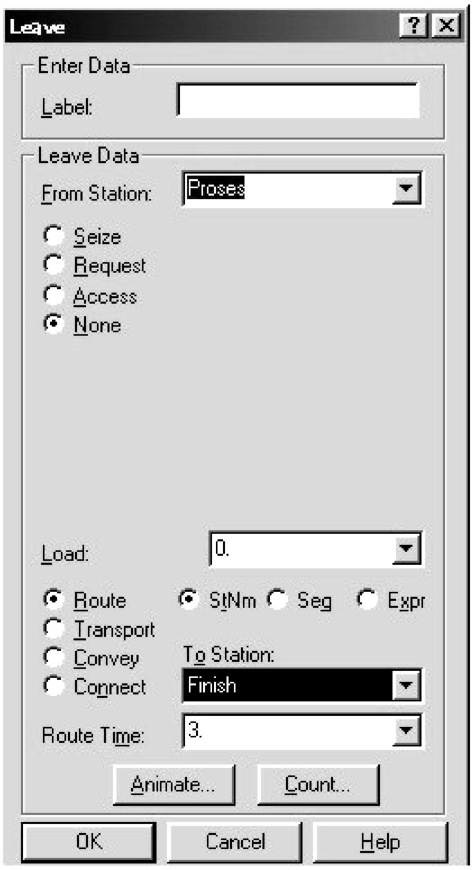
Capacity: 1

☑ Resource Statistics

Process Time: EXPO(6)

Next Label:

5. Module Leave – Pelanggan Meninggalkan Stasiun Pelayanan



Leave

Enter Data

Label:

Leave Data

From Station: Proses

○ Seize
○ Request
○ Access
● None

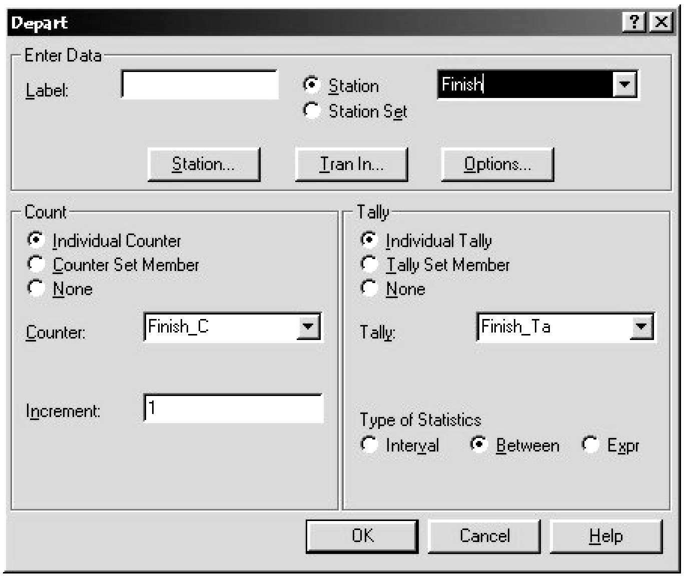
Load: 0.

● Route ● StNm ○ Seg ○ Expr
○ Transport
○ Convey
○ Connect

To Station: Finish

Route Time: 3.

6. Module Depart – Pelanggan Meninggalkan Sistem



Depart

Enter Data

Label:

● Station
○ Station Set

Finish

Count

● Individual Counter
○ Counter Set Member
○ None

Counter: Finish_C

Increment: 1

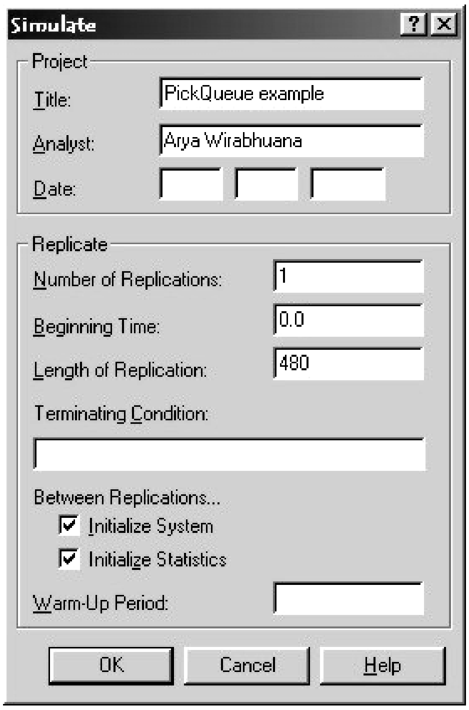
Tally

● Individual Tally
○ Tally Set Member
○ None

Tally: Finish_Ta

Type of Statistics
○ Interval ● Between ○ Expr

7. Module Simulate – Menjalankan Simulasi



Simulate

Project

Title: PickQueue example

Analyst: Arya Wirabhuana

Date:

Replicate

Number of Replications: 1

Beginning Time: 0.0

Length of Replication: 480

Terminating Condition:

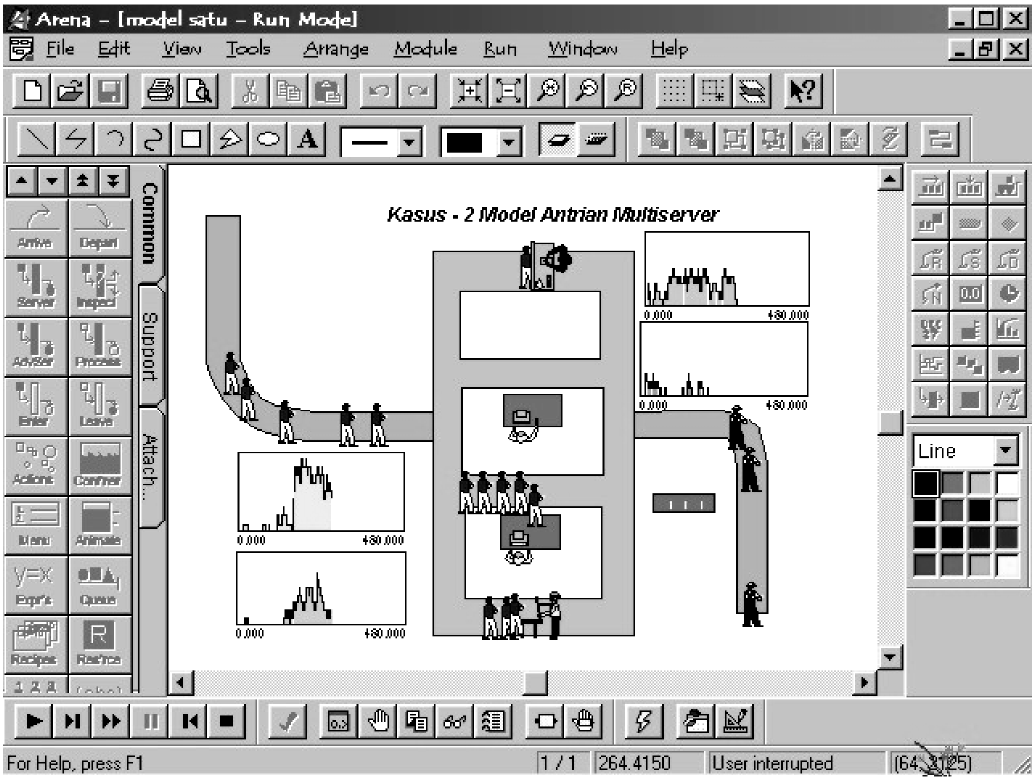
Between Replications...

☑ Initialize System

☑ Initialize Statistics

Warm-Up Period:

Berikut ini adalah Model setelah dilakukan Animasi :



Arena – [model satu – Run Mode]

Kasus - 2 Model Antrian Multiserver

For Help, press F1 | 1 / 1 | 264.4150 | User interrupted | (64, 2125)

.3. *Report Summary/Hasil Simulasi*

Setelah segala sesuatu tentang pembuatan model selesai, maka kita dapat menjalankan simulasi untuk mendapatkan hasil sebagai berikut :

```
Summary for Replication 1 of 1
Project:  Model Antrian Mu                     Run execution date :   8/ 9/1999
Analyst:  Arya Wirabhuana                      Model revision date:   8/ 9/1999
Replication ended at time      : 480.0

                                 TALLY VARIABLES
Identifier              Average   Half Width  Minimum    Maximum   Observations
_______________________________________________________________________________

kasir_3_Q Queue Time    39.879     (Insuf)    .00000     87.643         74
kasir_4_Q Queue Time    13.613     (Insuf)    .00000     36.285         85
Finish_Ta               1.4063     .16430     .00000     8.4515        333

                           DISCRETE-CHANGE VARIABLES
Identifier              Average   Half Width  Minimum    Maximum   Final Value
_______________________________________________________________________________

kasir_2 Available       1.0000     (Insuf)    1.0000     1.0000     1.0000
kasir_1 Available       1.0000     (Insuf)    1.0000     1.0000     1.0000
# in kasir_4_Q          2.4206     (Insuf)    .00000     8.0000     1.0000
# in kasir_3_Q          7.6903     (Insuf)    .00000     21.000     15.000
# in kasir_2_Q          .49472     (Insuf)    .00000     3.0000     3.0000
kasir_4 Busy            .81345     (Insuf)    .00000     1.0000     1.0000
# in kasir_1_Q          2.3489     (Insuf)    .00000     5.0000     .00000
kasir_3 Busy            .91255     (Insuf)    .00000     1.0000     1.0000
kasir_2 Busy            .63352     (Insuf)    .00000     1.0000     1.0000
kasir_1 Busy            .98245     (Insuf)    .00000     1.0000     1.0000
kasir_4 Available       1.0000     (Insuf)    1.0000     1.0000     1.0000
kasir_3 Available       1.0000     (Insuf)    1.0000     1.0000     1.0000

                                   COUNTERS
                    Identifier                Count   Limit
                    _________________________________________

                    Finish_C                    334  Infinite

Simulation run time: 1.70 minutes.
Simulation run complete.
```

Chapter 1
What is Simulation?

By : Arya Wirabhuana

Department Of Industrial Engineering

UIN SUNAN KALIJAGA YOGYAKARTA

Simulation Is ...

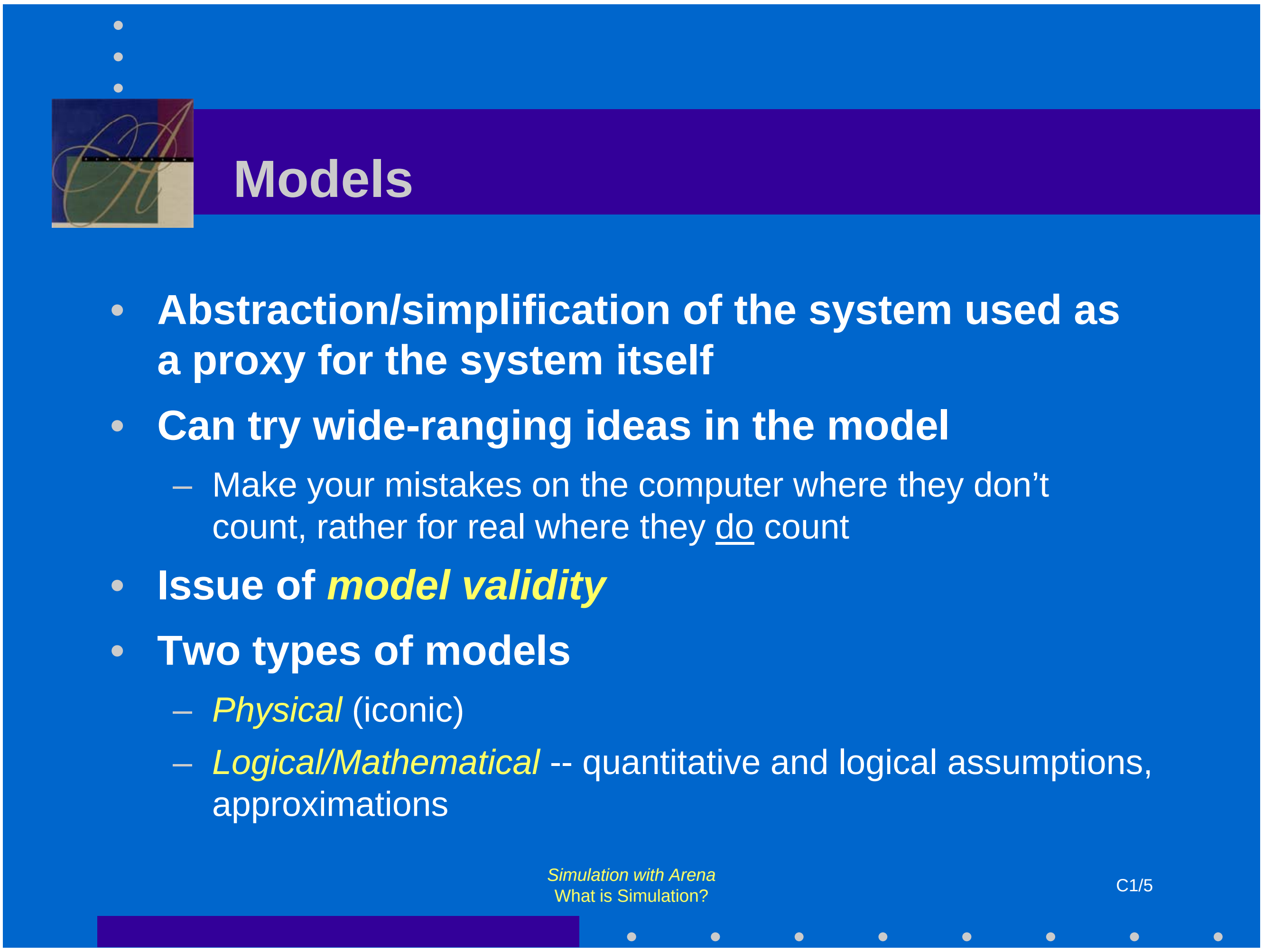
- **Very broad term, set of problems/approaches**
- **Generally, imitation of a system via computer**
- **Involves a model—validity?**
- **Don’t even aspire to analytic solution**
  - Don’t get exact results (bad)
  - Allows for complex, realistic models (good)
- **Approximate answer to exact problem is better than exact answer to approximate problem**
- **Consistently ranked as most useful, powerful of mathematical-modeling approaches**

Some Application Areas

- **Manufacturing—scheduling, inventory**
- **Staffing personal-service operations**
  - Banks, fast food, theme parks, Post Office, ...
- **Distribution and logistics**
- **Health care—emergency, operating rooms**
- **Computer systems**
- **Telecommunications**
- **Military**
- **Public policy**
  - Emergency planning
  - Courts, prisons, probation/parole

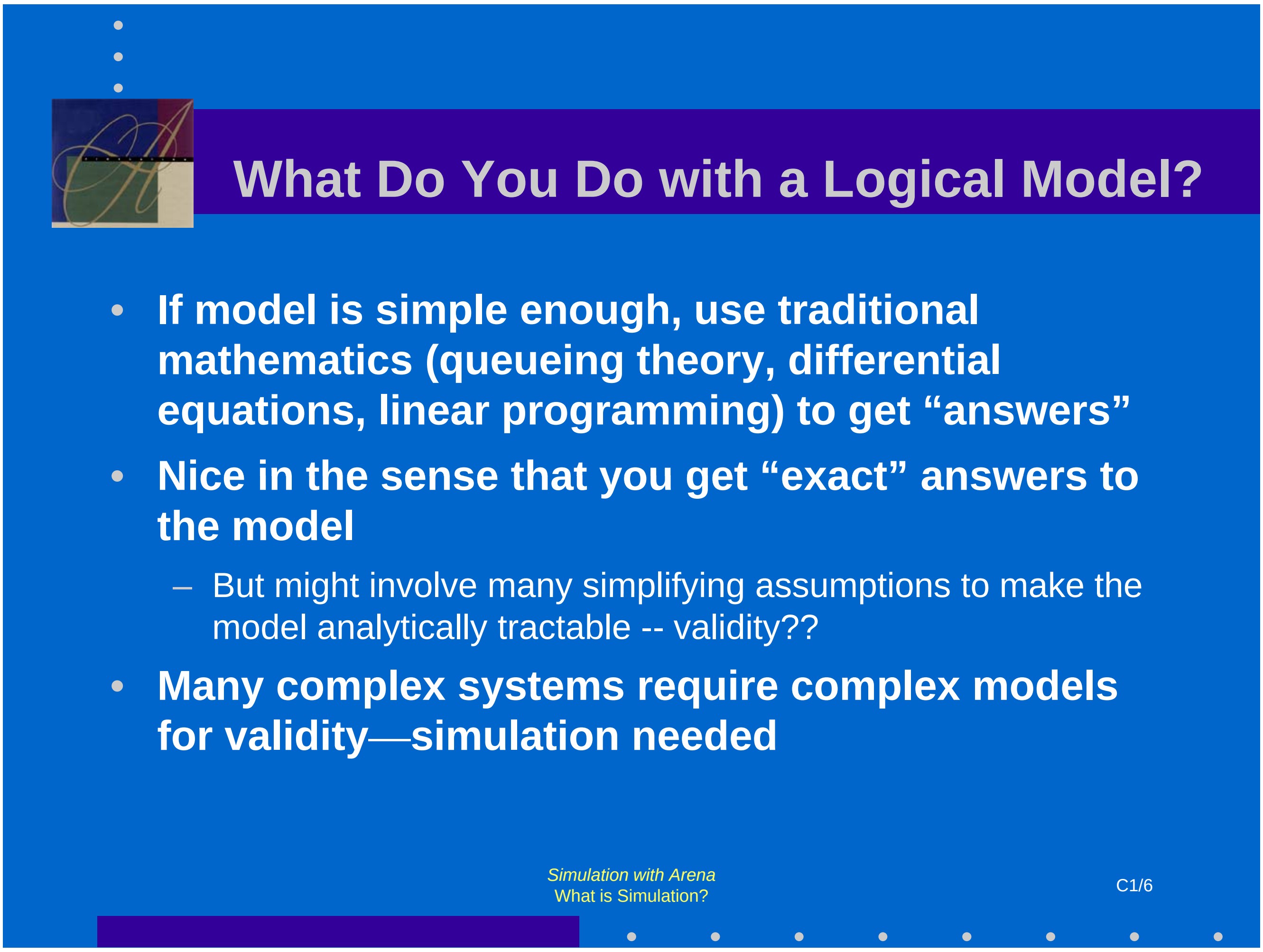
Systems

- **Physical facility/process, actual or planned**
- **Study its performance**
  - Measure
  - Improve
  - Design (if it doesn’t exist)
  - Maybe control in real time
- **Sometimes possible to “play” with the system**
- **But sometimes impossible to do so**
  - Doesn’t exist
  - Disruptive, expensive



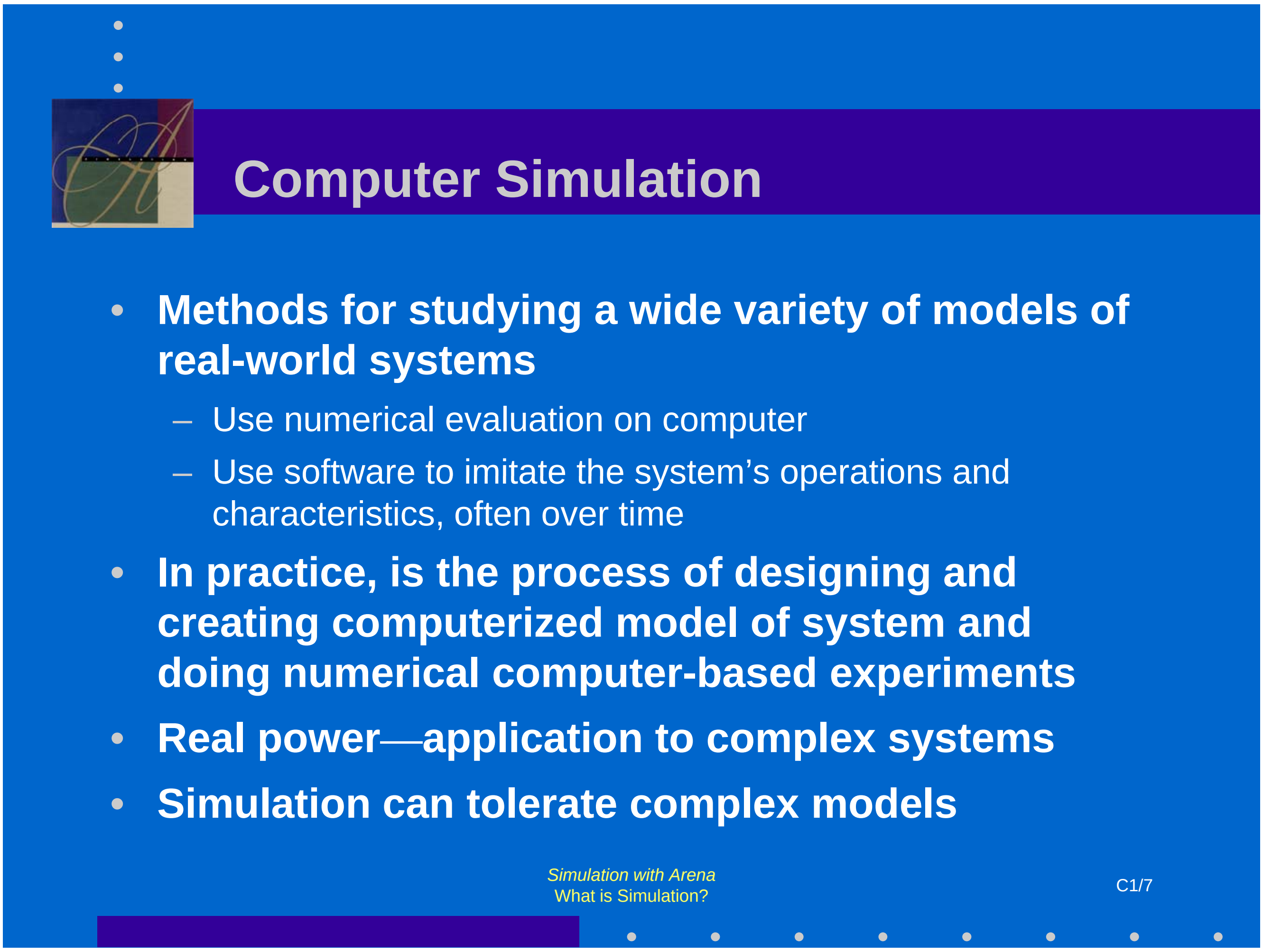
Models

- **Abstraction/simplification of the system used as a proxy for the system itself**
- **Can try wide-ranging ideas in the model**
  - Make your mistakes on the computer where they don’t count, rather for real where they do count
- **Issue of *model validity***
- **Two types of models**
  - *Physical* (iconic)
  - *Logical/Mathematical* -- quantitative and logical assumptions, approximations



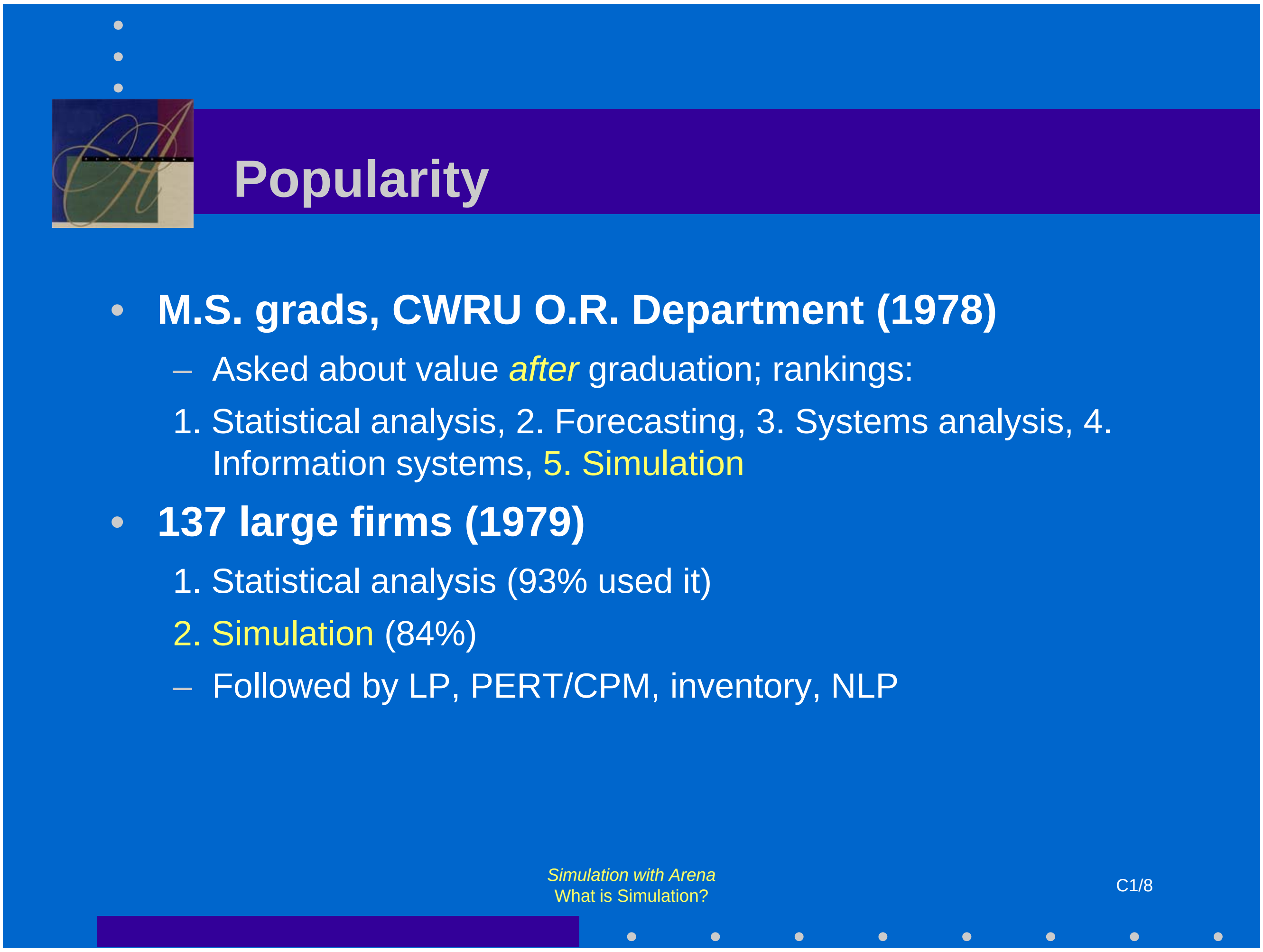
What Do You Do with a Logical Model?

- **If model is simple enough, use traditional mathematics (queueing theory, differential equations, linear programming) to get “answers”**
- **Nice in the sense that you get “exact” answers to the model**
  - But might involve many simplifying assumptions to make the model analytically tractable -- validity??
- **Many complex systems require complex models for validity—simulation needed**



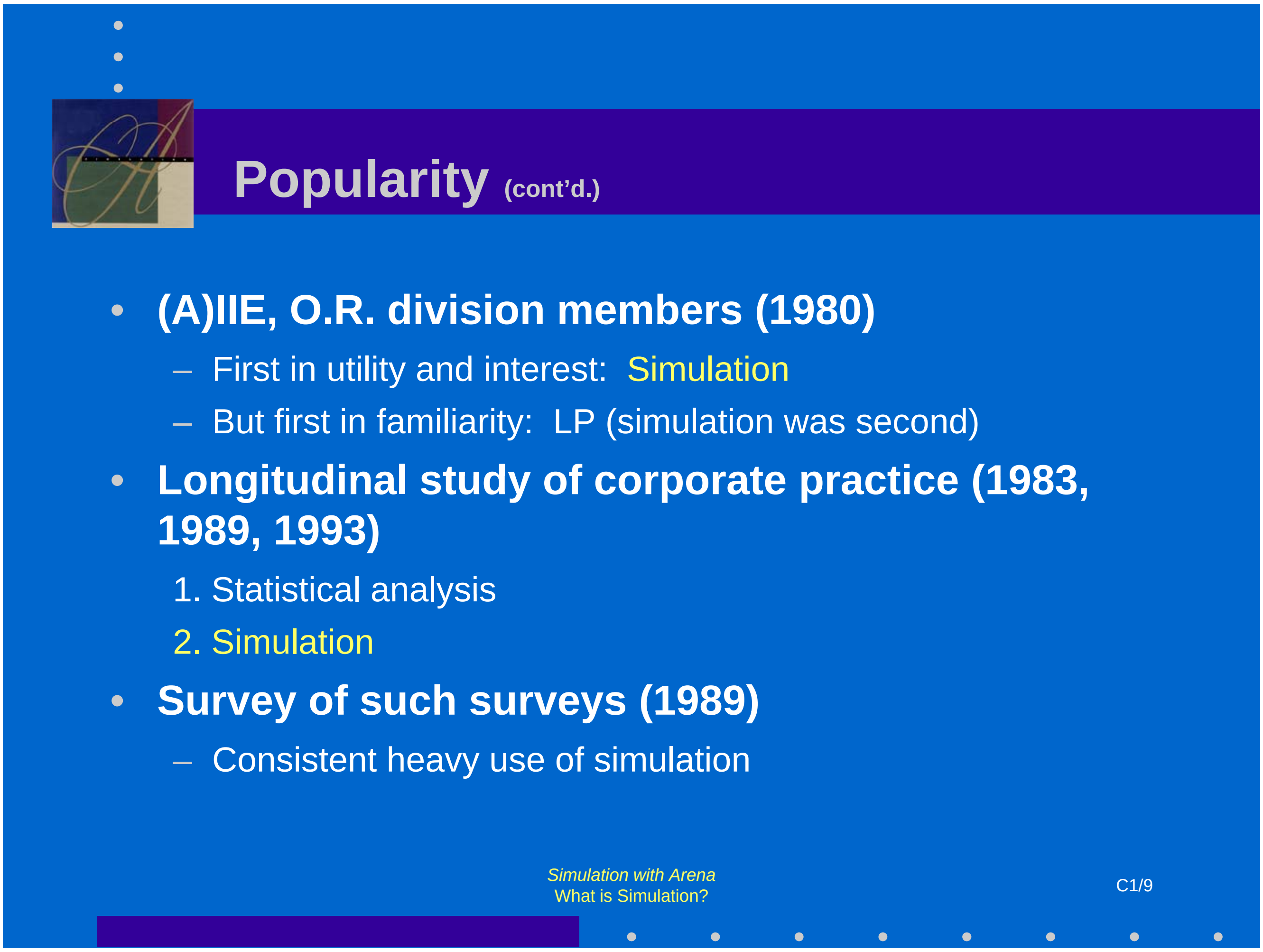
Computer Simulation

- **Methods for studying a wide variety of models of real-world systems**
  - Use numerical evaluation on computer
  - Use software to imitate the system’s operations and characteristics, often over time
- **In practice, is the process of designing and creating computerized model of system and doing numerical computer-based experiments**
- **Real power—application to complex systems**
- **Simulation can tolerate complex models**



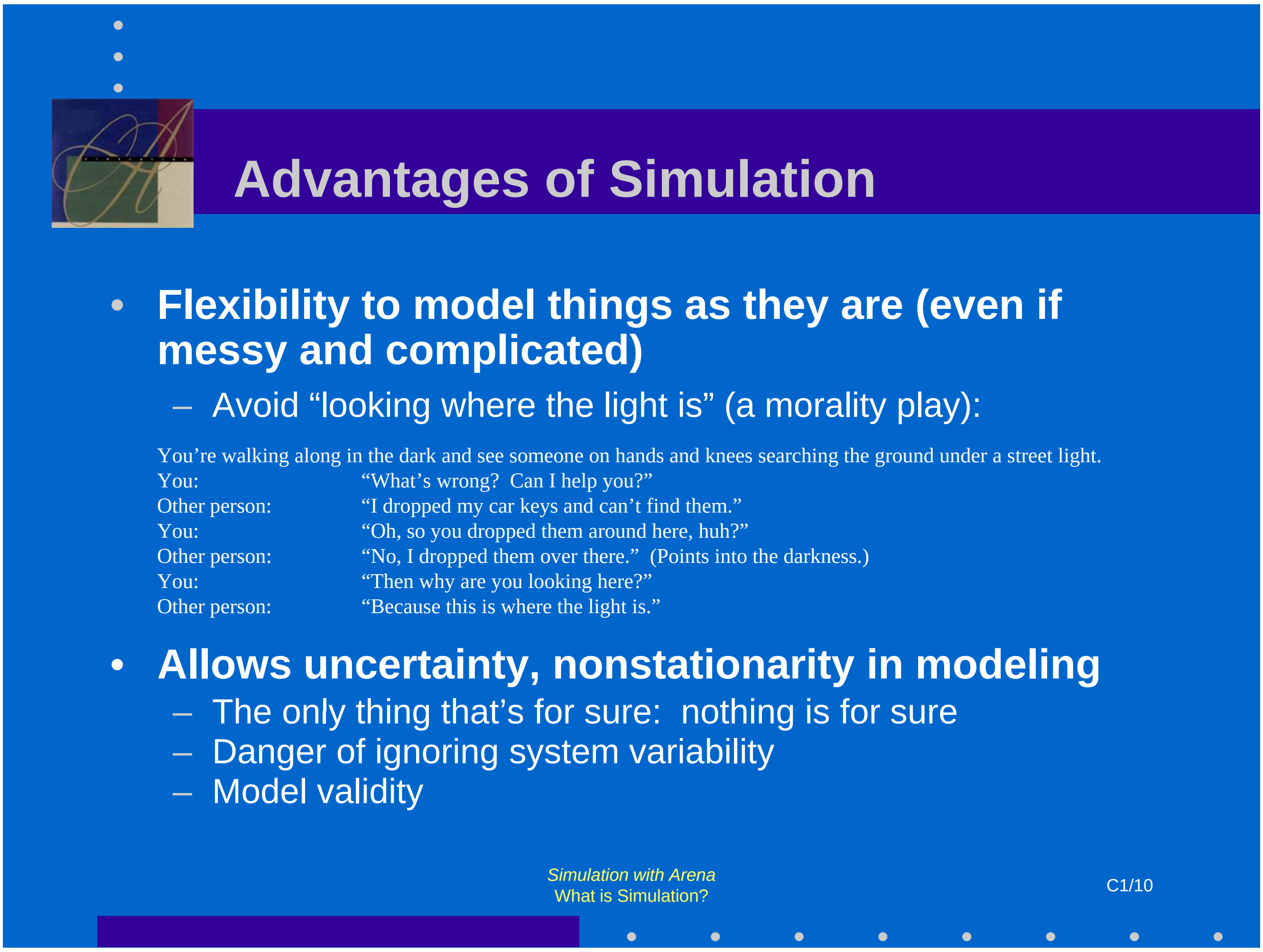
Popularity

- **M.S. grads, CWRU O.R. Department (1978)**
  - Asked about value *after* graduation; rankings:
  1. Statistical analysis, 2. Forecasting, 3. Systems analysis, 4. Information systems, 5. Simulation
- **137 large firms (1979)**
  1. Statistical analysis (93% used it)
  2. Simulation (84%)
  - Followed by LP, PERT/CPM, inventory, NLP



Popularity (cont'd.)

- **(A)IIE, O.R. division members (1980)**
  - First in utility and interest: Simulation
  - But first in familiarity: LP (simulation was second)
- **Longitudinal study of corporate practice (1983, 1989, 1993)**
  1. Statistical analysis
  2. Simulation
- **Survey of such surveys (1989)**
  - Consistent heavy use of simulation



Advantages of Simulation

- **Flexibility to model things as they are (even if messy and complicated)**
  - Avoid “looking where the light is” (a morality play):

You’re walking along in the dark and see someone on hands and knees searching the ground under a street light.
You: “What’s wrong? Can I help you?”
Other person: “I dropped my car keys and can’t find them.”
You: “Oh, so you dropped them around here, huh?”
Other person: “No, I dropped them over there.” (Points into the darkness.)
You: “Then why are you looking here?”
Other person: “Because this is where the light is.”

- **Allows uncertainty, nonstationarity in modeling**
  - The only thing that’s for sure: nothing is for sure
  - Danger of ignoring system variability
  - Model validity

Advantages of Simulation (cont'd.)

- **Advances in computing/cost ratios**
  - Estimated that 75% of computing power is used for various kinds of simulations
  - Dedicated machines (e.g., real-time shop-floor control)
- **Advances in simulation software**
  - Far easier to use (GUIs)
  - No longer as restrictive in modeling constructs (hierarchical, down to C)
  - Statistical design & analysis capabilities

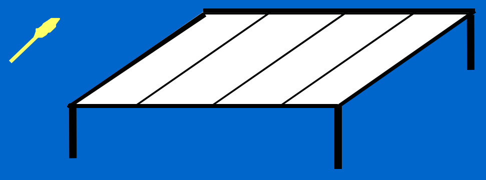
The Bad News

- **Don't get exact answers, only approximations, estimates**
  - Also true of many other modern methods
  - Can bound errors by machine roundoff
- **Get random output (*RIRO*) from stochastic simulations**
  - Statistical design, analysis of simulation experiments
  - Exploit: noise control, replicability, sequential sampling, variance-reduction techniques
  - Catch: "standard" statistical methods seldom work

Different Kinds of Simulation

- **Static vs. *Dynamic***
  - Does time have a role in the model?
- **Continuous-change vs. *Discrete-change***
  - Can the “state” change continuously or only at discrete points in time?
- **Deterministic vs. *Stochastic***
  - Is everything for sure or is there uncertainty?
- **Most operational models:**
  - *Dynamic*, *Discrete-change*, *Stochastic*

Simulation by Hand: The Buffon Needle Problem



- **Estimate π (George Louis Leclerc, c. 1733)**
- **Toss needle of length *l* onto table with stripes *d* (>*l*) apart**
- ***P* (needle crosses a line) =** $\frac{2l}{\pi d}$
- **Repeat; tally** $\hat{p}$ **= proportion of times a line is crossed**
- **Estimate π by** $\frac{2l}{\hat{p}d}$

Why Toss Needles?

- **Buffon needle problem seems silly now, but it has important simulation features:**
  - Experiment to *estimate* something hard to compute exactly (in 1733)
  - *Randomness*, so estimate will not be exact; estimate the error in the estimate
  - *Replication* (the more the better) to reduce error
  - *Sequential sampling* to control error -- keep tossing until probable error in estimate is “small enough”
  - *Variance reduction* (Buffon Cross)

Using Computers to Simulate

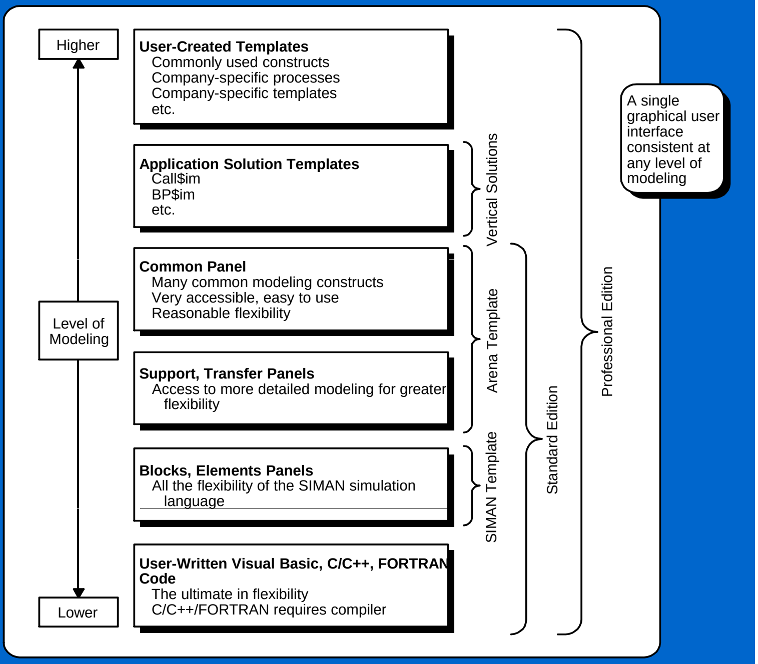
- **General-purpose languages (FORTRAN)**
  - Tedious, low-level, error-prone
  - But, almost complete flexibility
- **Support packages**
  - Subroutines for list processing, bookkeeping, time advance
  - Widely distributed, widely modified
- **Spreadsheets**
  - Usually static models
  - Financial scenarios, distribution sampling, SQC

Using Computers to Simulate (cont'd.)

- **Simulation languages**
  - GPSS, SIMSCRIPT, SLAM, SIMAN
  - Popular, in wide use today
  - Learning curve for features, effective use, syntax
- **High-level simulators**
  - Very easy, graphical interface
  - Domain-restricted (manufacturing, communications)
  - Limited flexibility—model validity?

Where Arena Fits In

- **Hierarchical structure**
  - Multiple levels of modeling
  - Can mix different modeling levels together in the same model
  - Often, start high then go lower as needed
- **Get ease-of-use advantage of simulators without sacrificing modeling flexibility**



User-Created Templates
Commonly used constructs
Company-specific processes
Company-specific templates
etc.

Application Solution Templates
Call$im
BP$im
etc.

Common Panel
Many common modeling constructs
Very accessible, easy to use
Reasonable flexibility

Support, Transfer Panels
Access to more detailed modeling for greater flexibility

Blocks, Elements Panels
All the flexibility of the SIMAN simulation language

User-Written Visual Basic, C/C++, FORTRAN Code
The ultimate in flexibility
C/C++/FORTRAN requires compiler

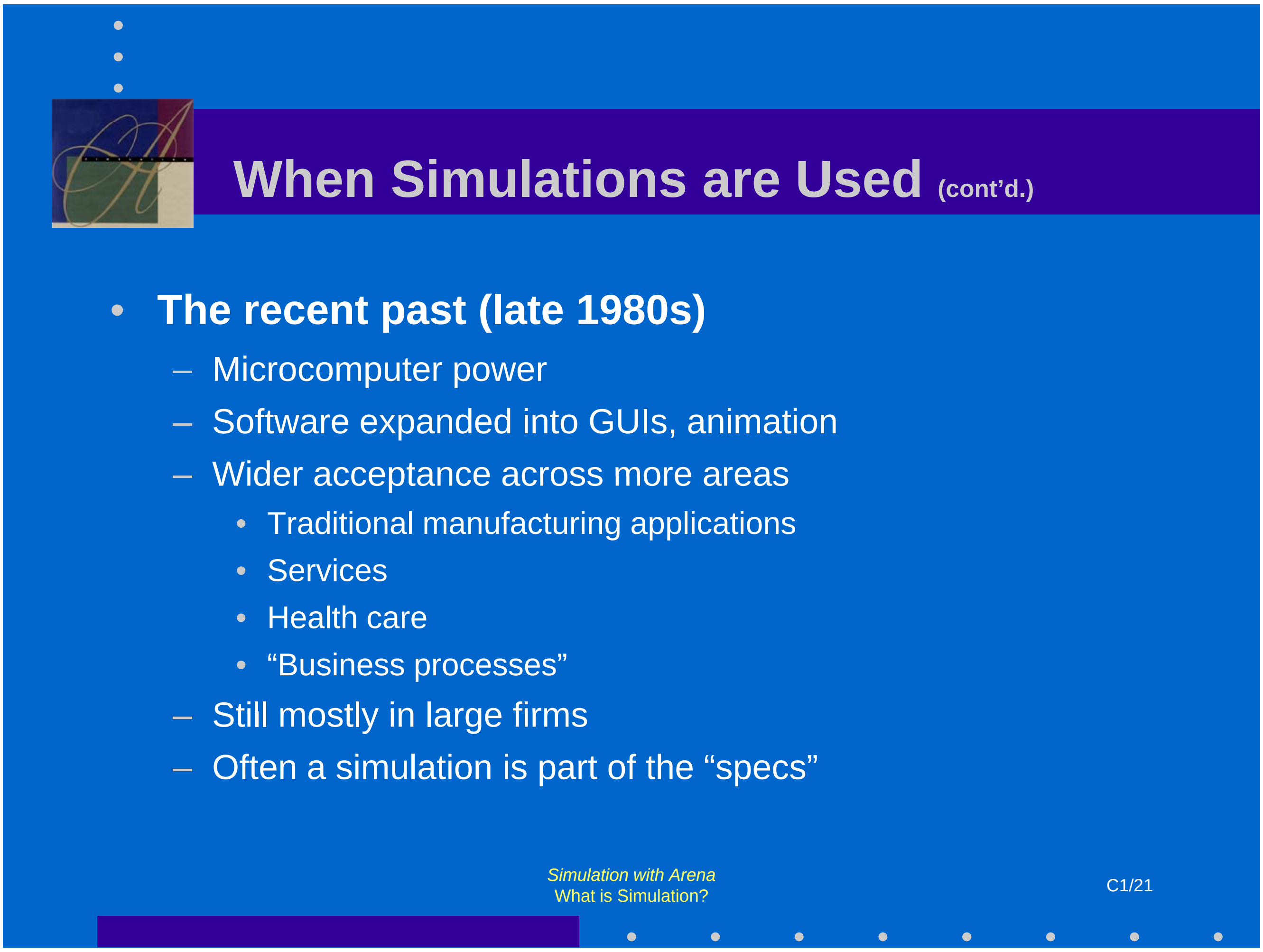
A single graphical user interface consistent at any level of modeling

When Simulations are Used

- **Uses of simulation have evolved with hardware, software**
- **The early years (1950s-1960s)**
  - Very expensive, specialized tool to use
  - Required big computers, special training
  - Mostly in FORTRAN (or even Assembler)
  - Processing cost as high as $1000/hour for a sub-286 level machine

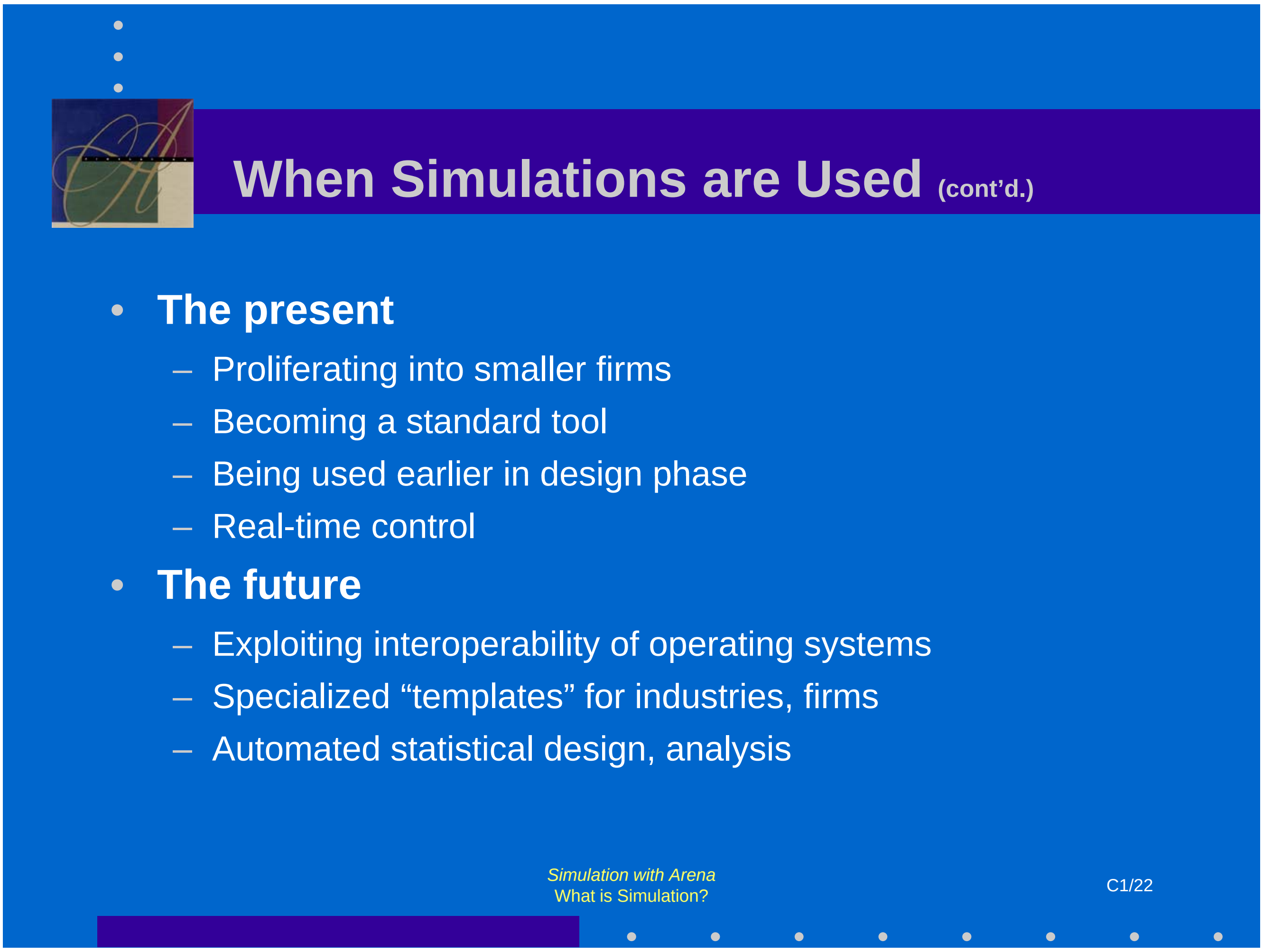
When Simulations are Used (cont'd.)

- **The formative years (1970s-early 1980s)**
  - Computers got faster, cheaper
  - Value of simulation more widely recognized
  - Simulation software improved, but they were still languages to be learned, typed, batch processed
  - Often used to clean up “disasters” in auto, aerospace industries
    - Car plant; heavy demand for certain model
    - Line underperforming
    - Simulated, problem identified
    - But demand had dried up—simulation was too late



When Simulations are Used (cont’d.)

- **The recent past (late 1980s)**
  - Microcomputer power
  - Software expanded into GUIs, animation
  - Wider acceptance across more areas
    - Traditional manufacturing applications
    - Services
    - Health care
    - “Business processes”
  - Still mostly in large firms
  - Often a simulation is part of the “specs”



When Simulations are Used (cont’d.)

- **The present**
  - Proliferating into smaller firms
  - Becoming a standard tool
  - Being used earlier in design phase
  - Real-time control
- **The future**
  - Exploiting interoperability of operating systems
  - Specialized “templates” for industries, firms
  - Automated statistical design, analysis

Chapter 2
Fundamental Simulation Concepts

By : Arya Wirabhuana

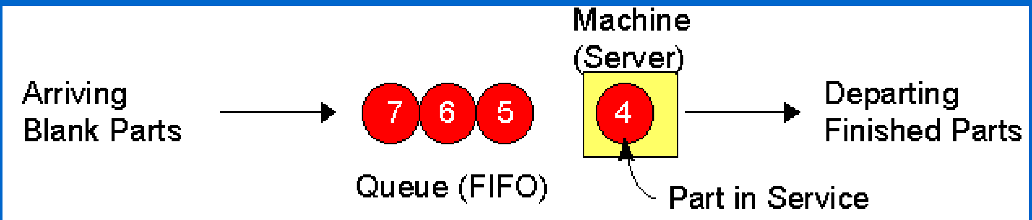
Department Of Industrial Engineering

UIN SUNAN KALIJAGA YOGYAKARTA

What We’ll Do ...

- **Underlying ideas, methods, and issues in simulation**
- **Software-independent (setting up for Arena)**
- **Centered around an example of a simple processing system**
  - Decompose the problem
  - Terminology
  - Simulation by hand
  - Some basic statistical issues
  - Overview of a simulation study

The System:
A Simple Processing System



- **General intent:**
  - Estimate expected production
  - Time in queue, queue length, proportion of time machine is busy
- **Telecommunications**
  - Be consistent (simulation doesn’t know)
  - Be reasonable (interpretation, roundoff error)

Model Specifics

- **Initially (time 0) empty and idle**
- **Time units: minutes**
- **Arrival times: 0.00, 6.84, 9.24, 11.94, 14.53**
  - *Interarrival* times: 6.84, 2.40, 2.70, 2.59, 0.73
- **Service times: 4.58, 2.96, 5.86, 3.21, 3.11**
- **Stop when 15 minutes of (simulated) time have passed**

Goals of the Study: Output Performance Measures

- ***Total production*** **of parts over the run (*P*)**
- ***Average waiting time*** **of parts in queue:**

$$\frac{\sum_{i=1}^{N} D_i}{N}$$

N = no. of parts completing queue wait
D_i = waiting time in queue of ith part
Know: $D_1 = 0$ (why?)
$N \geq 1$ (why?)

- ***Maximum waiting time*** **of parts in queue:**

$$\max_{i=1,\ldots,N} D_i$$

Goals of the Study: Output Performance Measures (cont'd.)

- ***Time-average number of parts*** **in queue:**

$$\frac{\int_0^{15} Q(t)\,dt}{15}$$

$Q(t)$ = number of parts in queue at time t

- ***Maximum number of parts*** **in queue:**

$$\max_{0 \le t \le 15} Q(t)$$

- ***Average*** **and** ***maximum flowtime*** **of parts (time in system, a.k.a.** ***cycle time*****)**

$$\sum_{i=1}^{P} F_i \Big/ P, \quad \max_{i=1,\ldots,P} F_i$$

F_i = flowtime of ith part

Goals of the Study: Output Performance Measures (cont'd.)

- *Utilization* of the machine (proportion of time busy)

$$\frac{\int_0^{15} B(t)\,dt}{15} \qquad B(t) = \begin{cases} 1 \text{ if the machine is busy at time } t \\ 0 \text{ if the machine is idle at time } t \end{cases}$$

- Many others possible (information overload?)

Analysis Options

- Educated guessing
  - Average interarrival time = 3.05 minutes
  - Average service time = 3.94 minutes
  - Model will "explode" eventually (but maybe not in 15 minutes)
  - If averages had come out for stable queue, assume all interarrivals, service times were equal to these values (no variability) -- never a queue!
  - Truth—between these extremes
  - Guessing has its limits

Analysis Options (cont'd.)

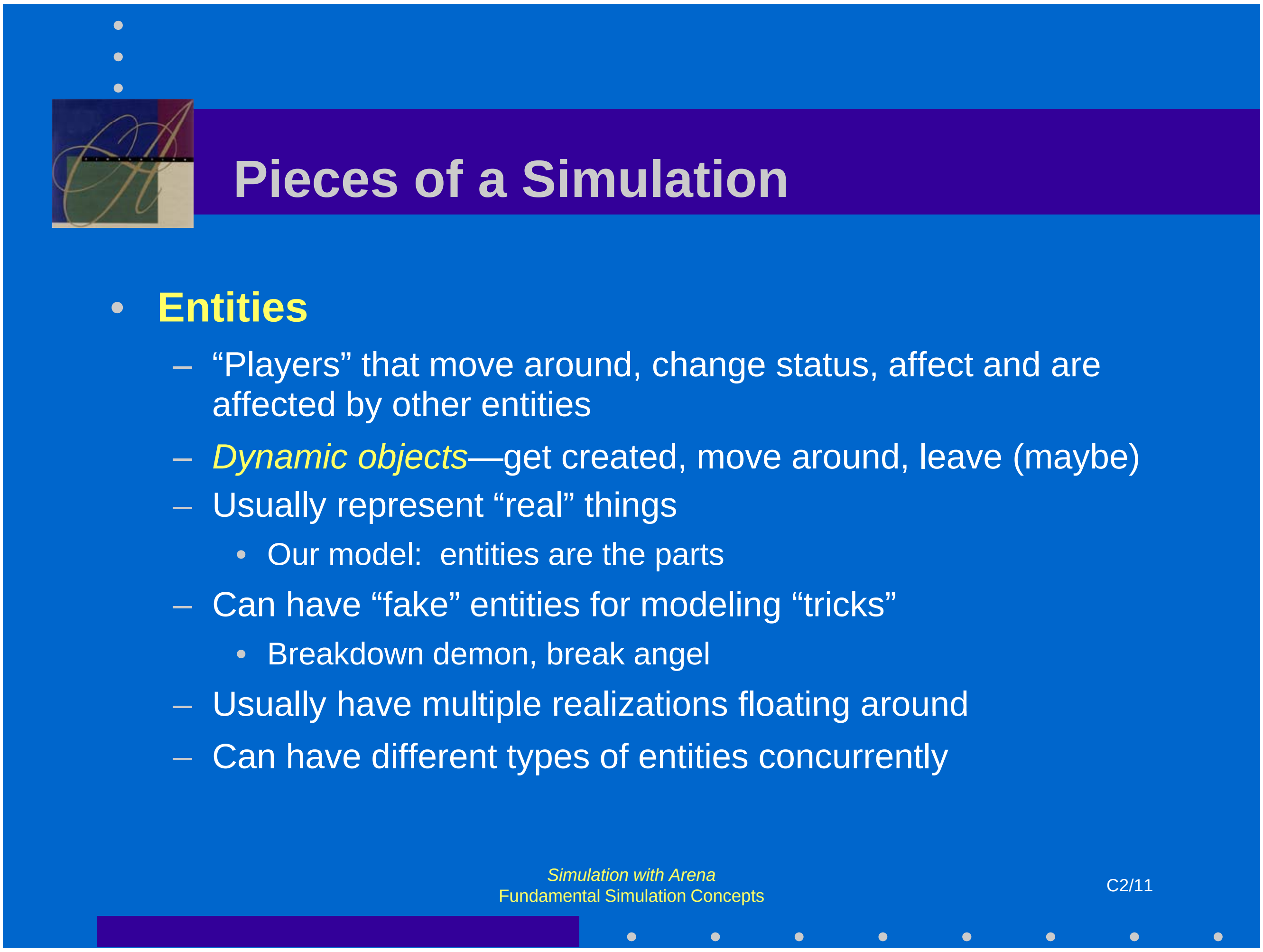
- **Queueing theory**
  - Requires additional assumptions about the model
  - Popular, simple model: *M/M/1 queue*
    - Interarrival times ~ exponential
    - Service times ~ exponential, independent of interarrivals
    - E(service) < E(interarrival)
    - Steady-state (long-run, forever)
    - Exact analytic results; e.g., avg. waiting time in queue is

$$\frac{\mu_S^2}{\mu_A - \mu_S} \qquad \begin{aligned} \mu_S &= E(\text{Service Time}) \\ \mu_A &= E(\text{Interarrival Time}) \end{aligned}$$

  - Problems: validity, estimating means, time frame
  - Often useful as first-cut approximation

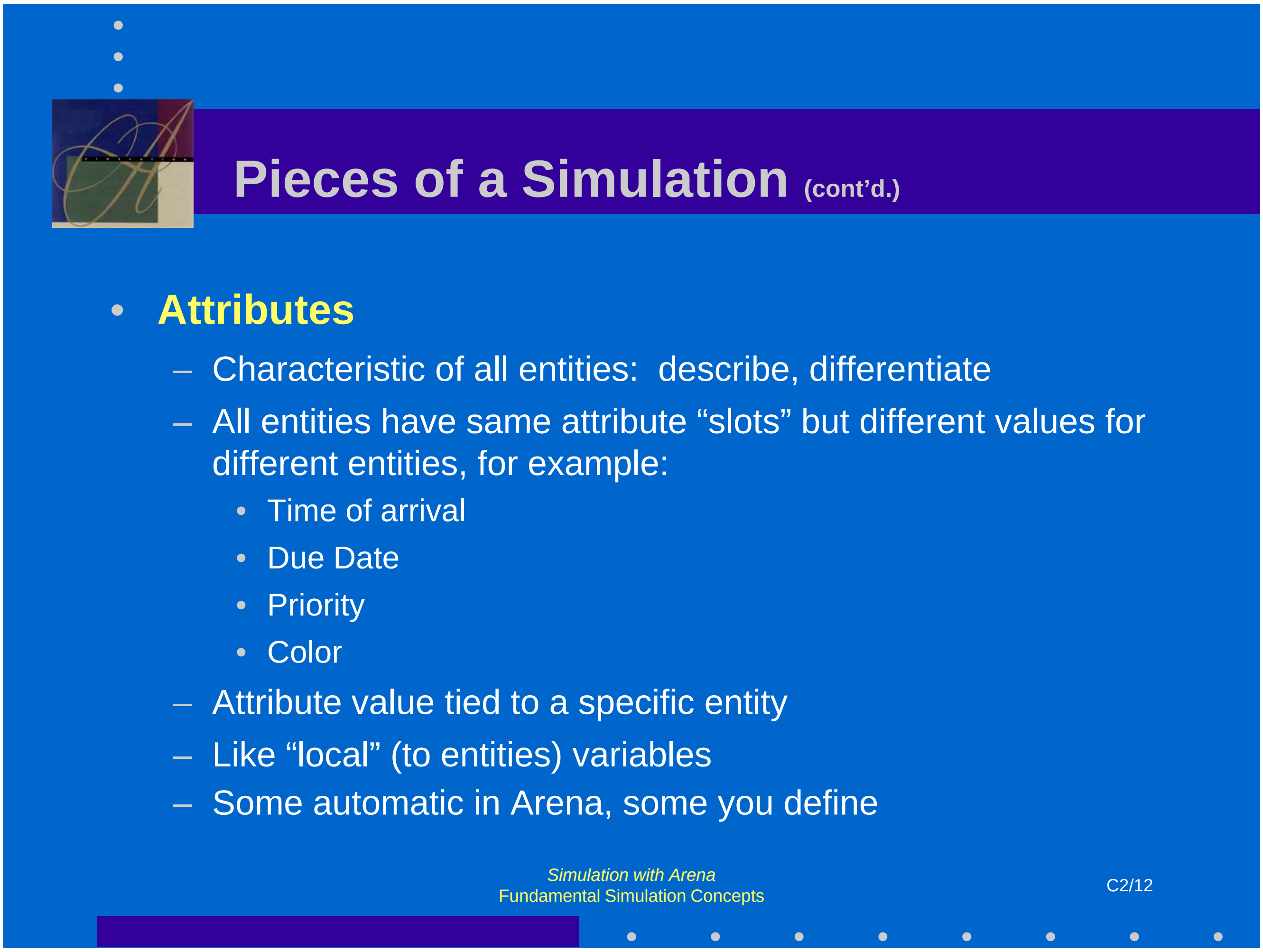
Mechanistic Simulation

- **Individual operations (arrivals, service times) will occur exactly as in reality**
- **Movements, changes occur at the right "time," in the right order**
- **Different pieces interact**
- **Install "observers" to get output performance measures**
- **Concrete, "brute-force" analysis approach**
- **Nothing mysterious or subtle**
  - But a lot of details, bookkeeping
  - Simulation software keeps track for you



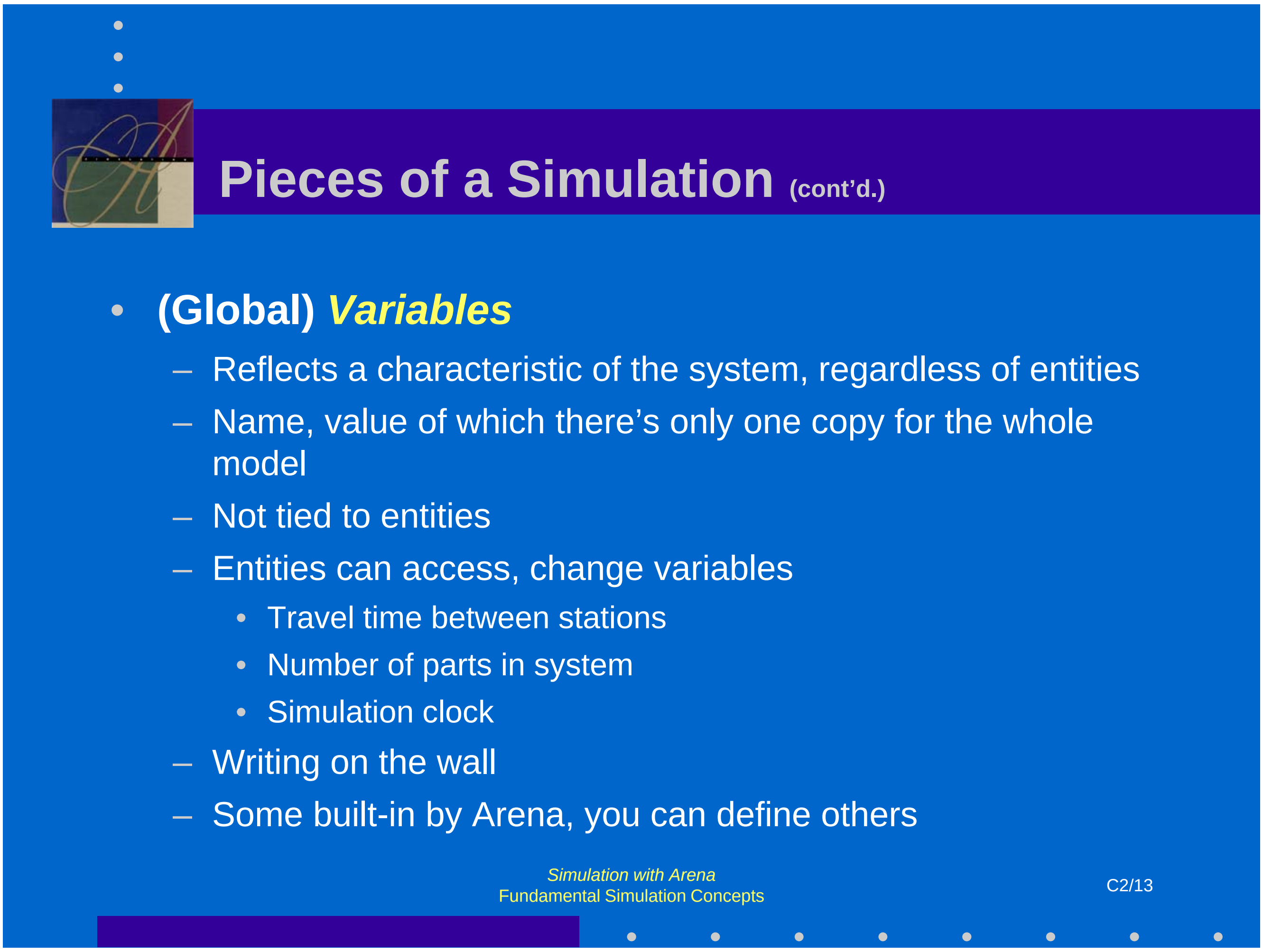
Pieces of a Simulation

- **Entities**
  - “Players” that move around, change status, affect and are affected by other entities
  - *Dynamic objects*—get created, move around, leave (maybe)
  - Usually represent “real” things
    - Our model: entities are the parts
  - Can have “fake” entities for modeling “tricks”
    - Breakdown demon, break angel
  - Usually have multiple realizations floating around
  - Can have different types of entities concurrently



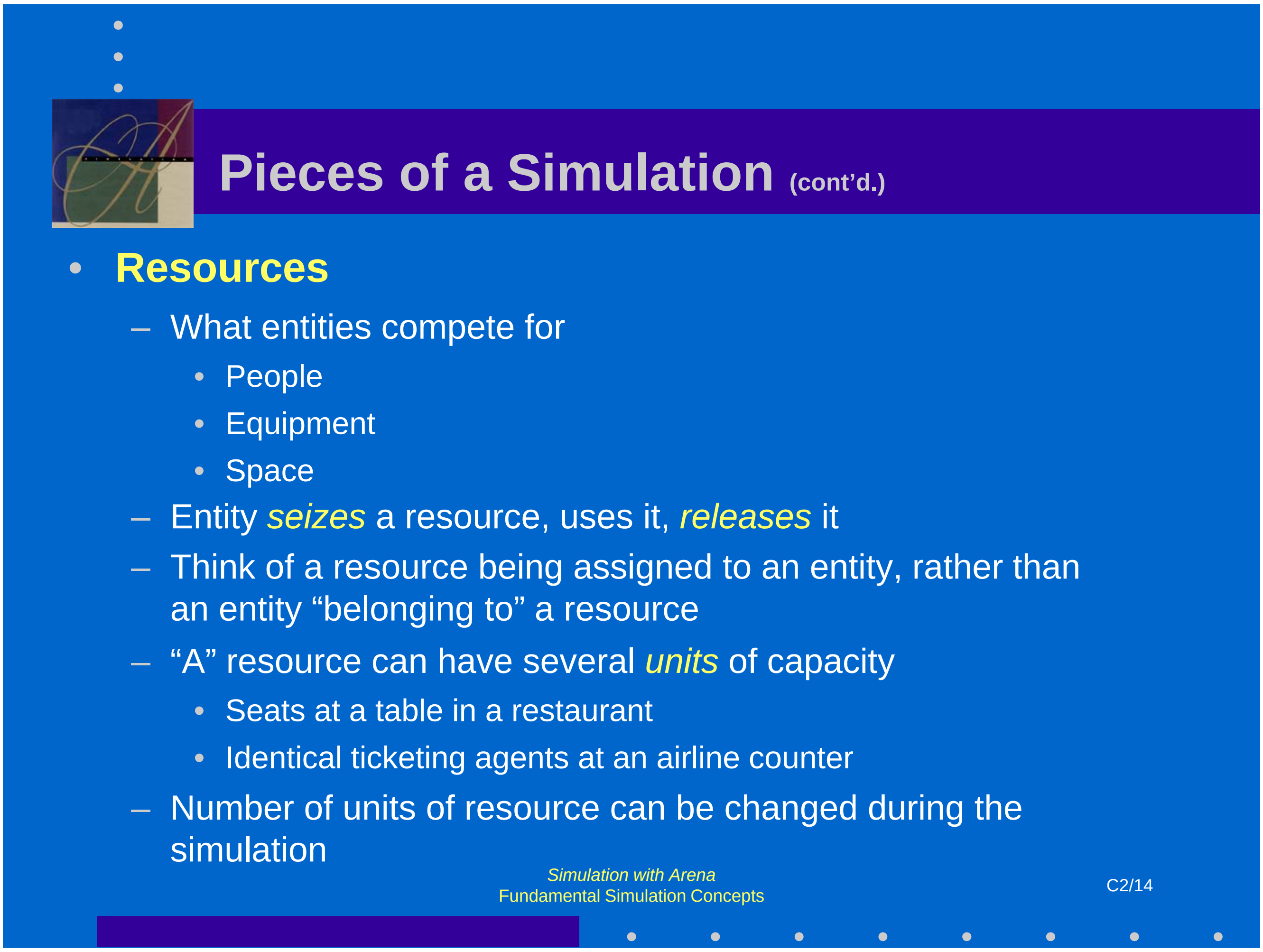
Pieces of a Simulation (cont’d.)

- **Attributes**
  - Characteristic of all entities: describe, differentiate
  - All entities have same attribute “slots” but different values for different entities, for example:
    - Time of arrival
    - Due Date
    - Priority
    - Color
  - Attribute value tied to a specific entity
  - Like “local” (to entities) variables
  - Some automatic in Arena, some you define



Pieces of a Simulation (cont'd.)

- (Global) ***Variables***
  - Reflects a characteristic of the system, regardless of entities
  - Name, value of which there's only one copy for the whole model
  - Not tied to entities
  - Entities can access, change variables
    - Travel time between stations
    - Number of parts in system
    - Simulation clock
  - Writing on the wall
  - Some built-in by Arena, you can define others



Pieces of a Simulation (cont'd.)

- **Resources**
  - What entities compete for
    - People
    - Equipment
    - Space
  - Entity *seizes* a resource, uses it, *releases* it
  - Think of a resource being assigned to an entity, rather than an entity "belonging to" a resource
  - "A" resource can have several *units* of capacity
    - Seats at a table in a restaurant
    - Identical ticketing agents at an airline counter
  - Number of units of resource can be changed during the simulation

Pieces of a Simulation (cont'd.)

- **Queues**
  - Place for entities to wait when they can't move on (maybe since the resource they want to seize is not available)
  - Have names, often tied to a corresponding resource
  - Can have a finite capacity to model limited space—have to model what to do if an entity shows up to a queue that's already full
  - Usually watch the length of a queue, waiting time in it

Pieces of a Simulation (cont'd.)

- **Statistical accumulators**
  - Variables that "watch" what's happening
  - Depend on output performance measures desired
  - "Passive" in model—don't participate, just watch
  - Many are automatic in Arena, but some you may have to set up and maintain during the simulation
  - At end of simulation, used to compute final output performance measures

Pieces of a Simulation (cont'd.)

- **Statistical accumulators for the simple processing system**
  - Number of parts produced so far
  - Total of the times spent in queue so far
  - No. of parts that have gone through the queue
  - Max time in queue we've seen
  - Total of flowtimes
  - Max flowtime we've seen
  - Area so far under queue-length curve $Q(t)$
  - Max of $Q(t)$ so far
  - Area so far under server-busy curve $B(t)$

Scheduling Dynamics: The Event-Scheduling World View

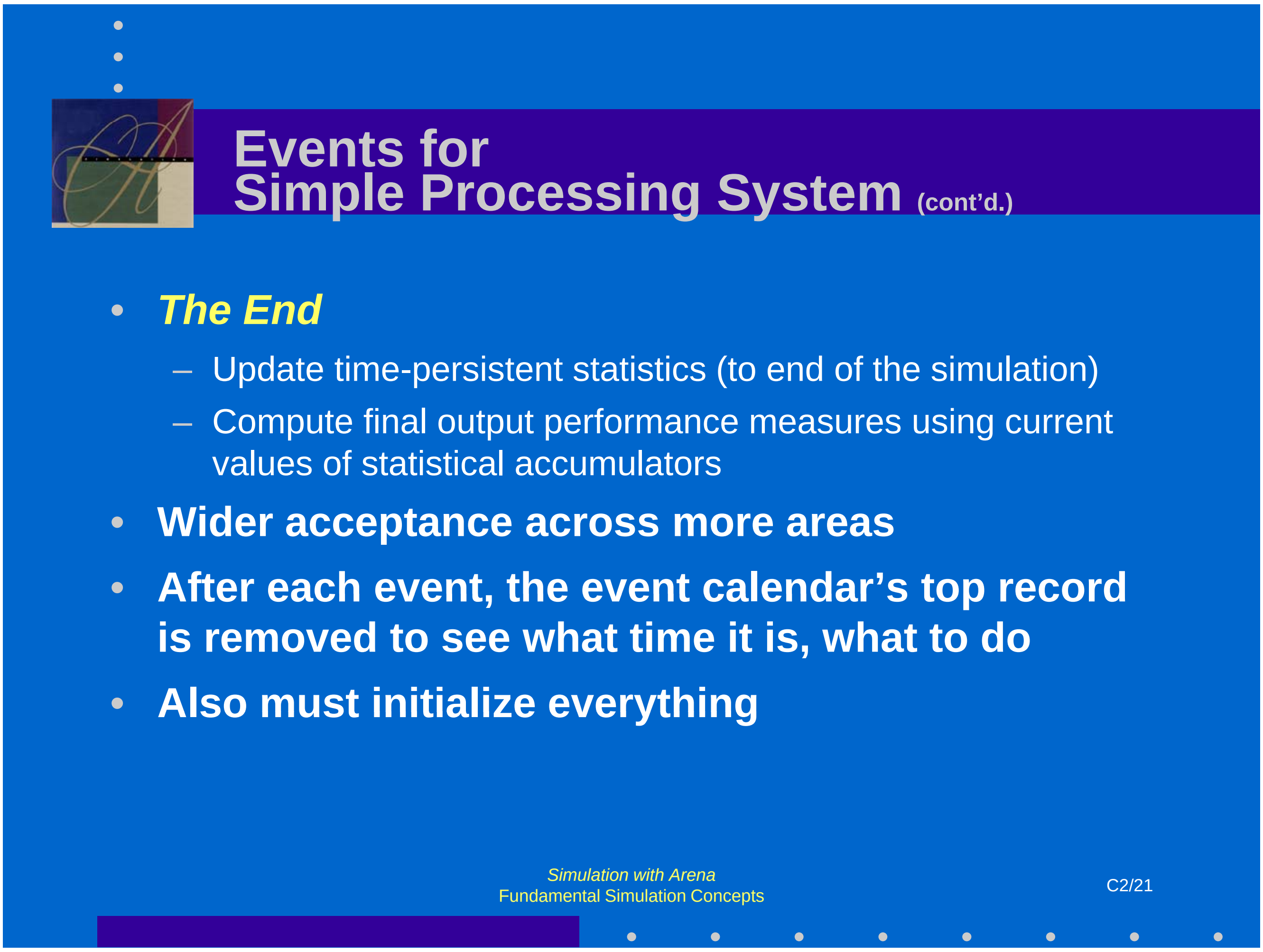
- **Identify characteristic *events*—change state**
- **Decide on *logic* to: effect *state changes* for each event type, observe statistics**
- **Keep a simulation *clock*, future *event calendar***
- ***Jump* from one event to the next, process, observe statistics, update event calendar**
- ***Stopping rule***
- **Usually done with general-purpose programming language (C, FORTRAN, etc.)**

Events for Simple Processing System

- ***Arrival*** **of new part to the system**
  - Update time-persistent statistical accumulators (from last event to now)
    - Area under *Q*(*t*)
    - Max of *Q*(*t*)
    - Area under *B*(*t*)
  - "Mark" arriving part with current time (use later)
  - If machine is idle:
    - Start processing (schedule departure), Make machine busy, Tally time in queue (0)
  - Else (machine is busy):
    - Put part at end of queue, Increase queue-length variable
  - Schedule the next arrival event

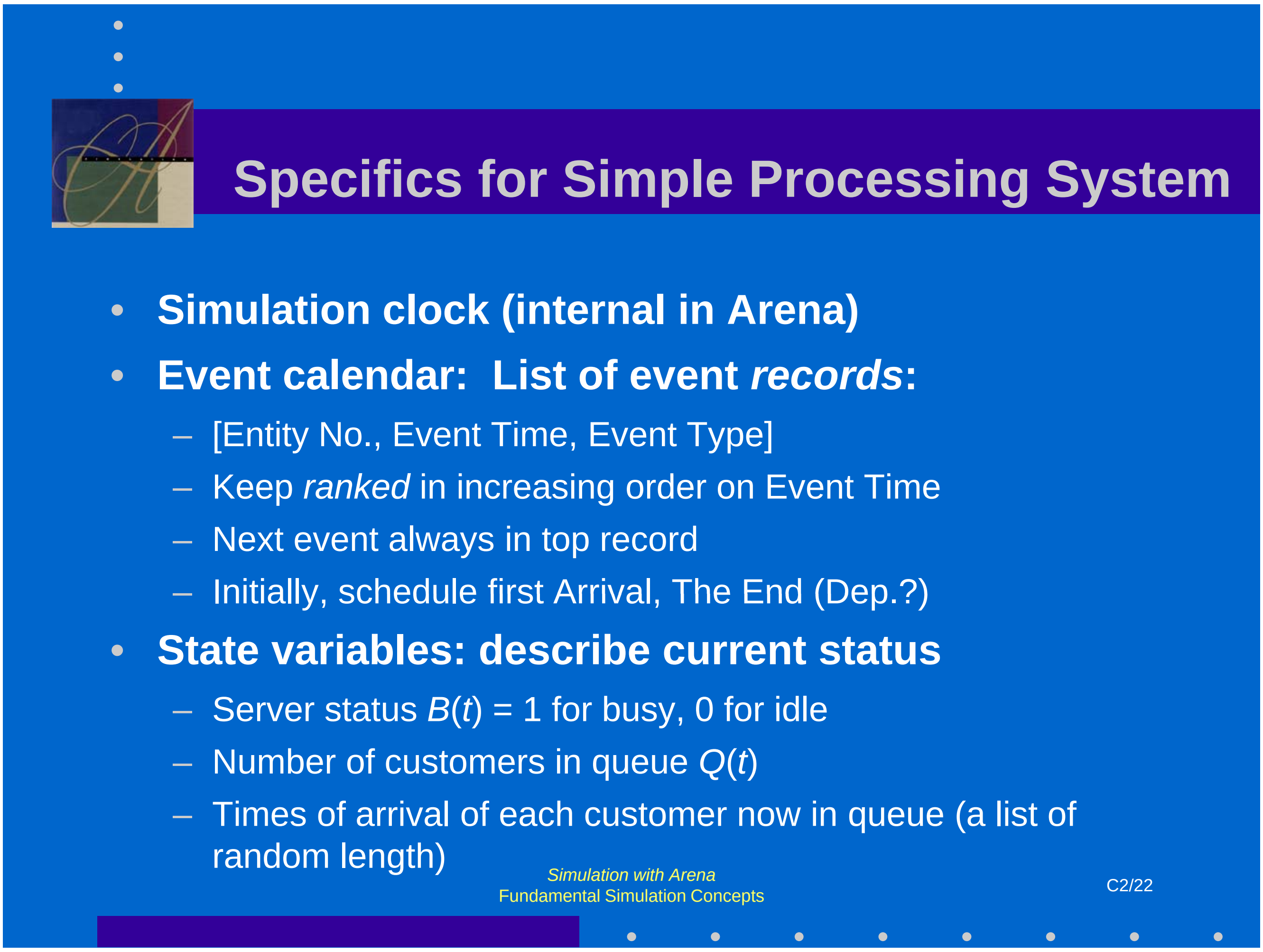
Events for Simple Processing System (cont'd.)

- ***Departure*** **(when a service is completed)**
  - Increment number-produced stat accumulator
  - Compute & tally *flowtime* (now - time of arrival)
  - Update time-persistent statistics
  - If queue is non-empty:
    - Take first part out of queue, compute & tally its time in queue, begin service (schedule departure event)
  - Else (queue is empty):
    - Make the machine idle (*Note*: there will be no departure event scheduled on the future events calendar)



Events for Simple Processing System (cont'd.)

- ***The End***
  - Update time-persistent statistics (to end of the simulation)
  - Compute final output performance measures using current values of statistical accumulators
- **Wider acceptance across more areas**
- **After each event, the event calendar's top record is removed to see what time it is, what to do**
- **Also must initialize everything**



Specifics for Simple Processing System

- **Simulation clock (internal in Arena)**
- **Event calendar: List of event *records*:**
  - [Entity No., Event Time, Event Type]
  - Keep *ranked* in increasing order on Event Time
  - Next event always in top record
  - Initially, schedule first Arrival, The End (Dep.?)
- **State variables: describe current status**
  - Server status $B(t)$ = 1 for busy, 0 for idle
  - Number of customers in queue $Q(t)$
  - Times of arrival of each customer now in queue (a list of random length)

Simulation by Hand

- Manually track state variables, statistical accumulators
- Use “given” interarrival, service times
- Keep track of event calendar
- “Lurch” clock from one event to the next
- Will omit flowtimes, “max” computations here (see text for complete details)

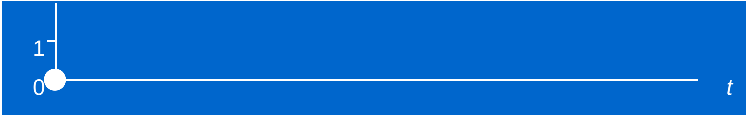
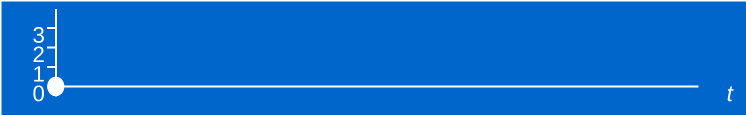
Simulation by Hand: Setup

<table>
<tr><td>System
🟨</td><td>Clock</td><td><i>B</i>(<i>t</i>)</td><td colspan="2"><i>Q</i>(<i>t</i>)</td><td>Arrival times of custs in queue</td><td colspan="5">Event calendar</td></tr>
<tr><td>No. of completed times in queue</td><td colspan="3">Total of times in queue</td><td colspan="2">Area under <i>B</i>(<i>t</i>)</td><td colspan="5">Area under <i>Q</i>(<i>t</i>)</td></tr>
<tr><td><i>B</i>(<i>t</i>) graph</td><td colspan="10"> </td></tr>
<tr><td><i>Q</i>(<i>t</i>) graph</td><td colspan="10"> </td></tr>
<tr><td>Interarrival times</td><td colspan="10">6.84 2.40 2.70 2.59 0.73</td></tr>
<tr><td>Service times</td><td colspan="10">4.91 3.09 6.45 3.40 3.25</td></tr>
</table>

Simulation by Hand: Initialize at *t* = 0.00

System	Clock	*B*(*t*)	*Q*(*t*)	Arrival times of custs in queue	Event calendar
	0.00	0	0	<empty>	[1, 0.00,Arr] [-, 15.00, End]

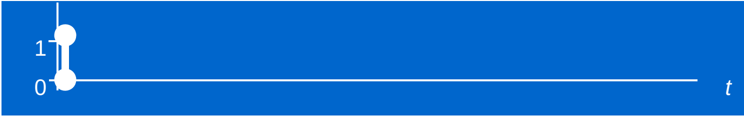
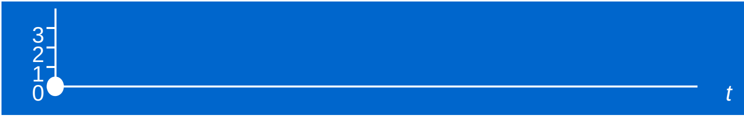
No. of completed times in queue	Total of times in queue	Area under *B*(*t*)	Area under *Q*(*t*)
0	0.00	0.00	0.00

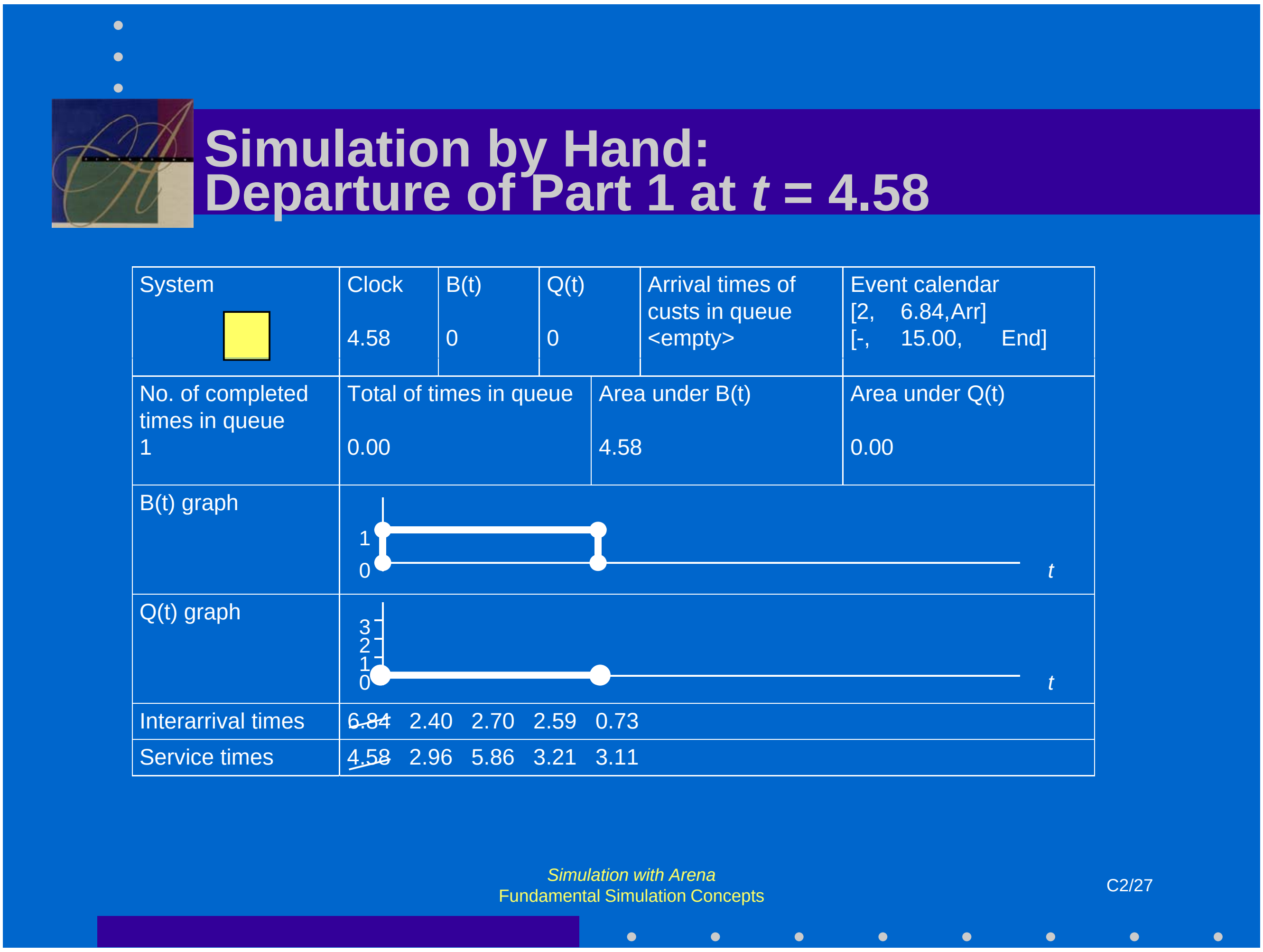
B(*t*) graph	
Q(*t*) graph	
Interarrival times	6.84 2.40 2.70 2.59 0.73
Service times	4.58 2.96 5.86 3.21 3.11

Simulation by Hand: Arrival of Part 1 at *t* = 0.00

System	Clock	*B*(*t*)	*Q*(*t*)	Arrival times of custs in queue	Event calendar
1	0.00	1	0	<empty>	[1, 4.58,Dep] [2, 6.84,Arr] [-, 15.00, End]

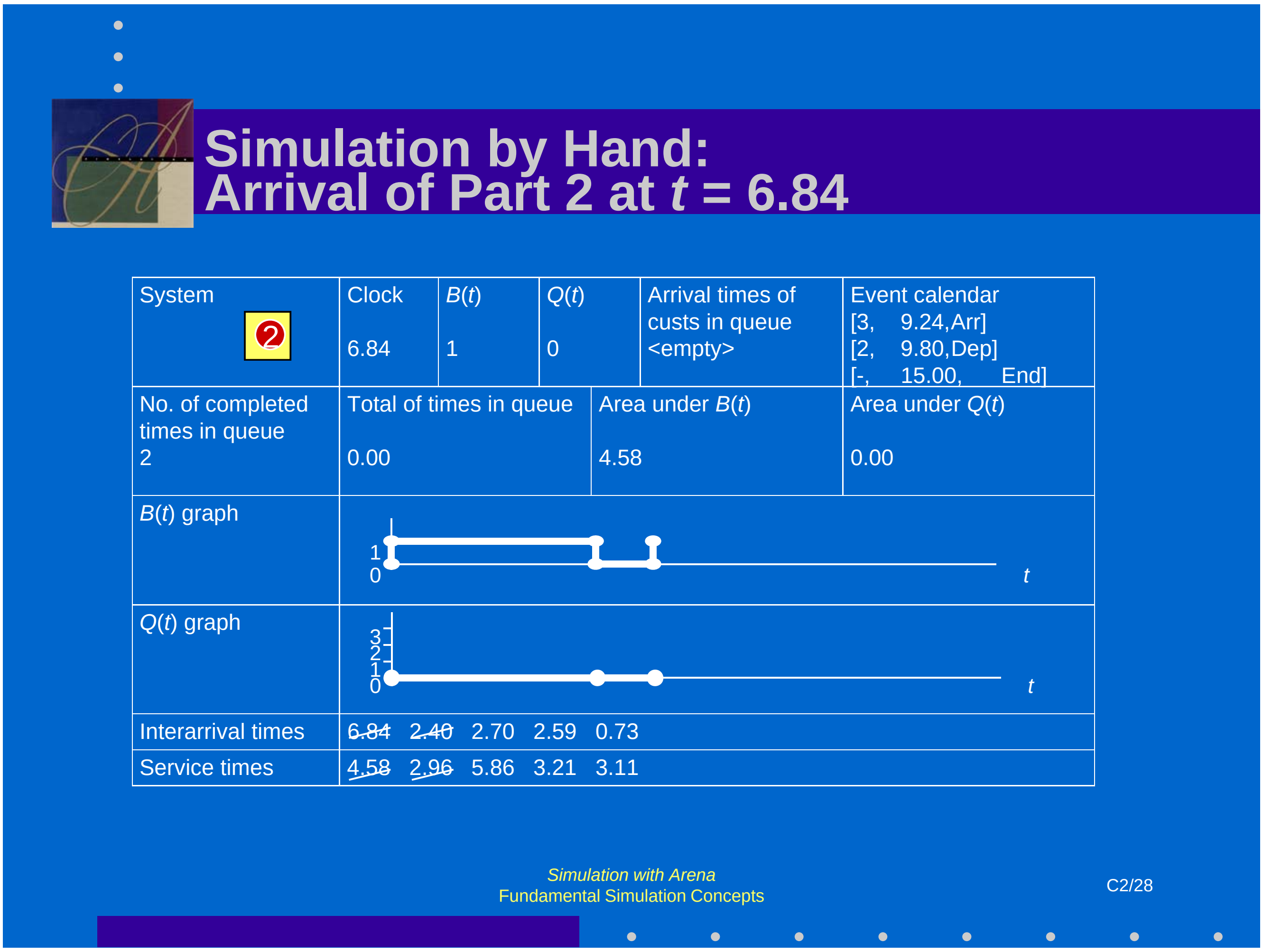
No. of completed times in queue	Total of times in queue	Area under *B*(*t*)	Area under *Q*(*t*)
1	0.00	0.00	0.00

B(*t*) graph	
Q(*t*) graph	
Interarrival times	~~6.84~~ 2.40 2.70 2.59 0.73
Service times	~~4.58~~ 2.96 5.86 3.21 3.11



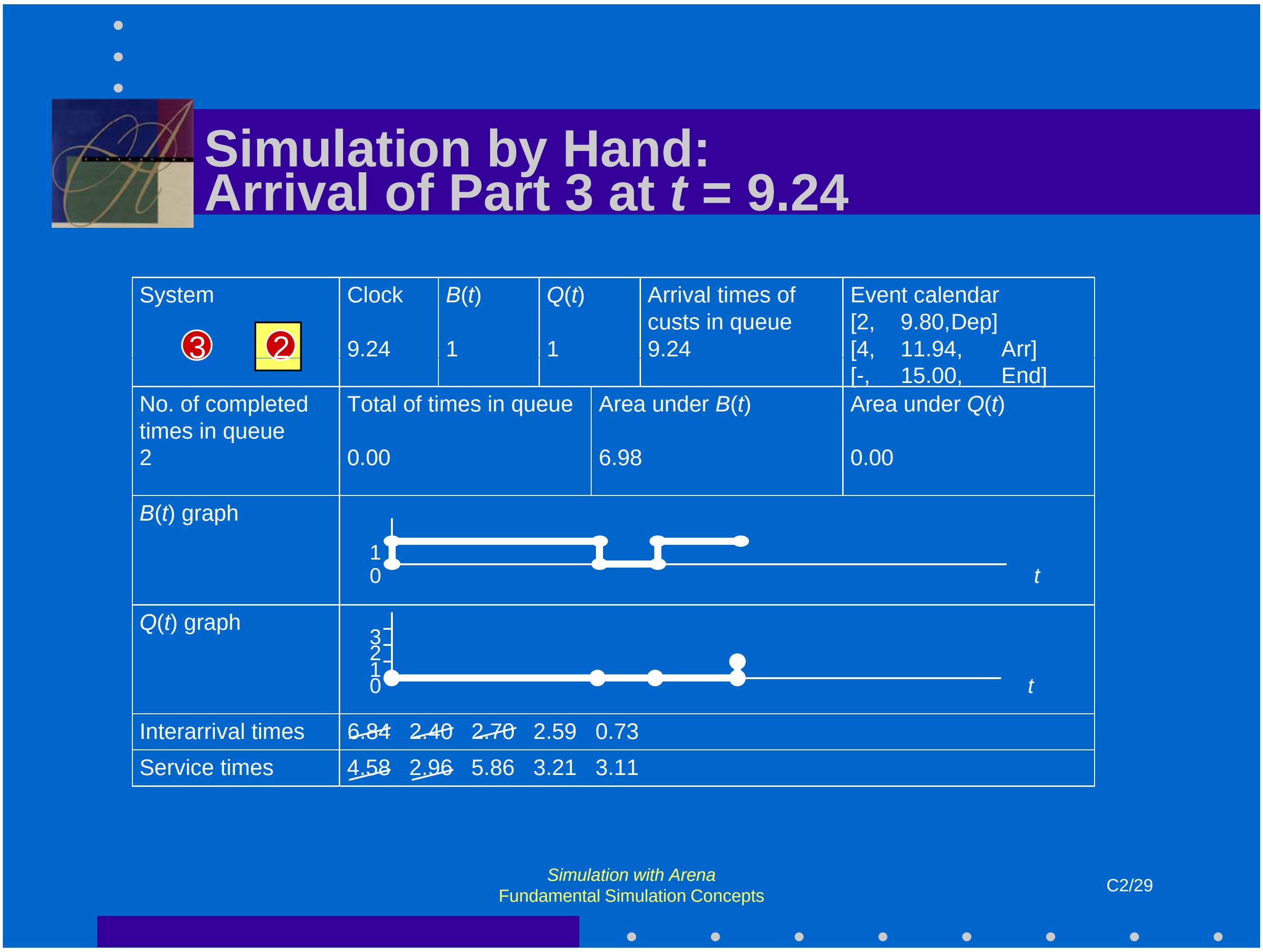
Simulation by Hand: Departure of Part 1 at *t* = 4.58

System	Clock 4.58	B(t) 0	Q(t) 0	Arrival times of custs in queue <empty>	Event calendar [2, 6.84,Arr] [-, 15.00, End]
No. of completed times in queue 1	Total of times in queue 0.00		Area under B(t) 4.58		Area under Q(t) 0.00
B(t) graph	1 0 *t*				
Q(t) graph	3 2 1 0 *t*				
Interarrival times	~~6.84~~ 2.40 2.70 2.59 0.73				
Service times	~~4.58~~ 2.96 5.86 3.21 3.11				



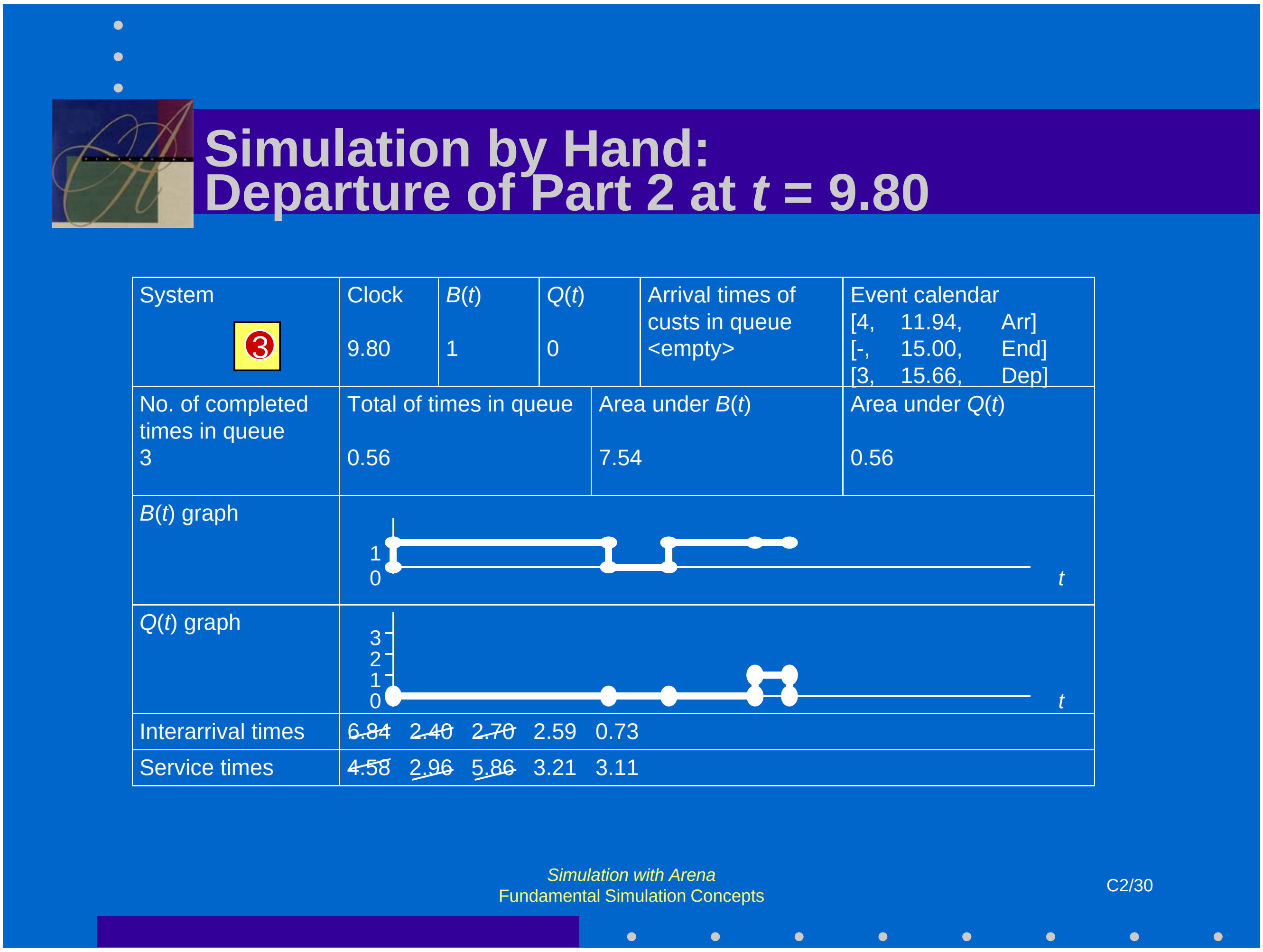
Simulation by Hand: Arrival of Part 2 at *t* = 6.84

System 2	Clock 6.84	*B*(*t*) 1	*Q*(*t*) 0	Arrival times of custs in queue <empty>	Event calendar [3, 9.24,Arr] [2, 9.80,Dep] [-, 15.00, End]
No. of completed times in queue 2	Total of times in queue 0.00		Area under *B*(*t*) 4.58		Area under *Q*(*t*) 0.00
B(*t*) graph	1 0 *t*				
Q(*t*) graph	3 2 1 0 *t*				
Interarrival times	~~6.84~~ ~~2.40~~ 2.70 2.59 0.73				
Service times	~~4.58~~ ~~2.96~~ 5.86 3.21 3.11				



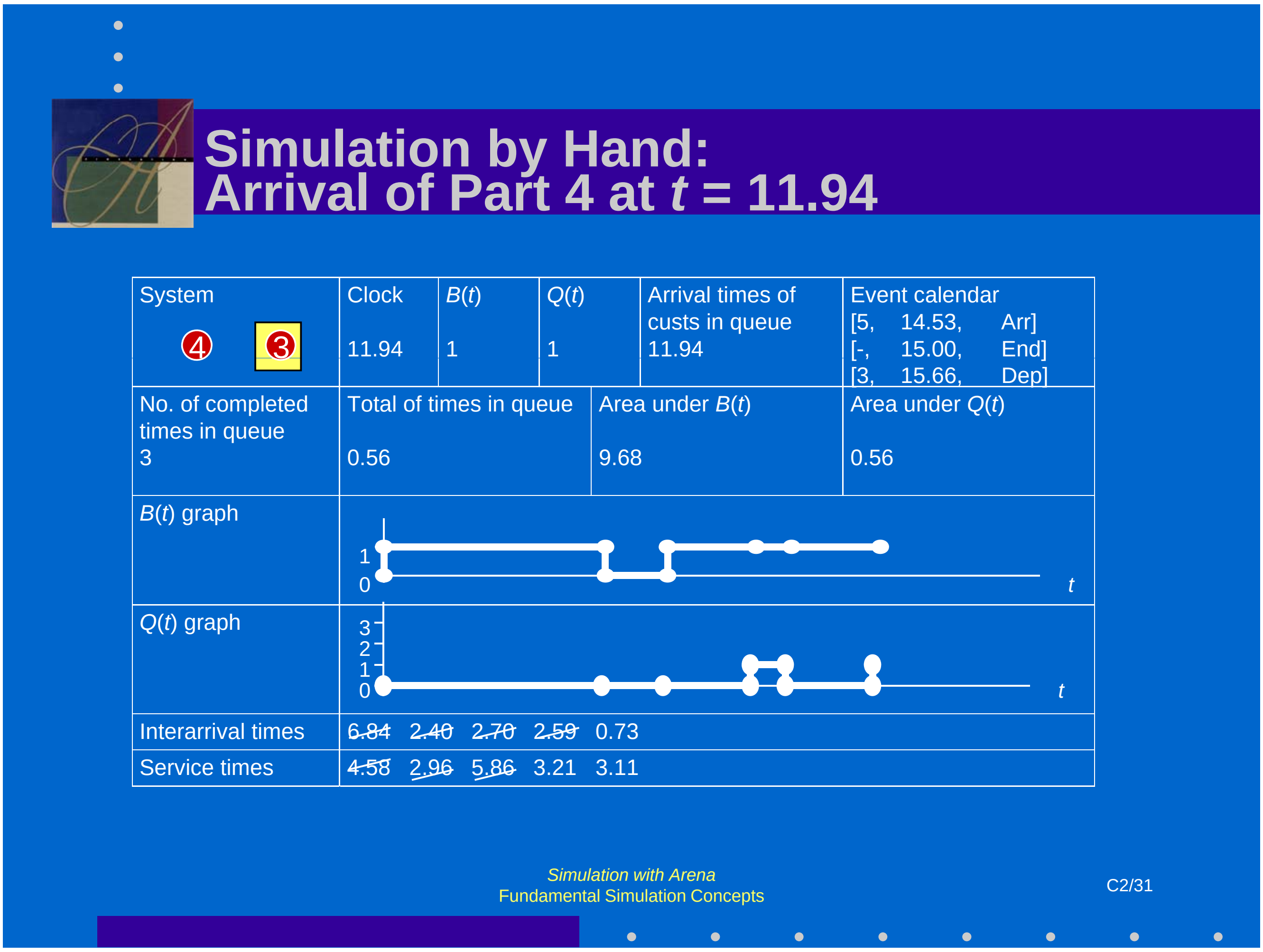
Simulation by Hand: Arrival of Part 3 at *t* = 9.24

System	Clock	*B*(*t*)	*Q*(*t*)	Arrival times of custs in queue	Event calendar
③ ②	9.24	1	1	9.24	[2, 9.80, Dep] [4, 11.94, Arr] [-, 15.00, End]
No. of completed times in queue 2	Total of times in queue 0.00		Area under *B*(*t*) 6.98		Area under *Q*(*t*) 0.00
B(*t*) graph	1 0 *t*				
Q(*t*) graph	3 2 1 0 *t*				
Interarrival times	~~6.84~~ ~~2.40~~ ~~2.70~~ 2.59 0.73				
Service times	~~4.58~~ ~~2.96~~ 5.86 3.21 3.11				



Simulation by Hand: Departure of Part 2 at *t* = 9.80

System	Clock	*B*(*t*)	*Q*(*t*)	Arrival times of custs in queue	Event calendar
③	9.80	1	0	<empty>	[4, 11.94, Arr] [-, 15.00, End] [3, 15.66, Dep]
No. of completed times in queue 3	Total of times in queue 0.56		Area under *B*(*t*) 7.54		Area under *Q*(*t*) 0.56
B(*t*) graph	1 0 *t*				
Q(*t*) graph	3 2 1 0 *t*				
Interarrival times	~~6.84~~ ~~2.40~~ ~~2.70~~ 2.59 0.73				
Service times	~~4.58~~ ~~2.96~~ ~~5.86~~ 3.21 3.11				



Simulation by Hand: Arrival of Part 4 at *t* = 11.94

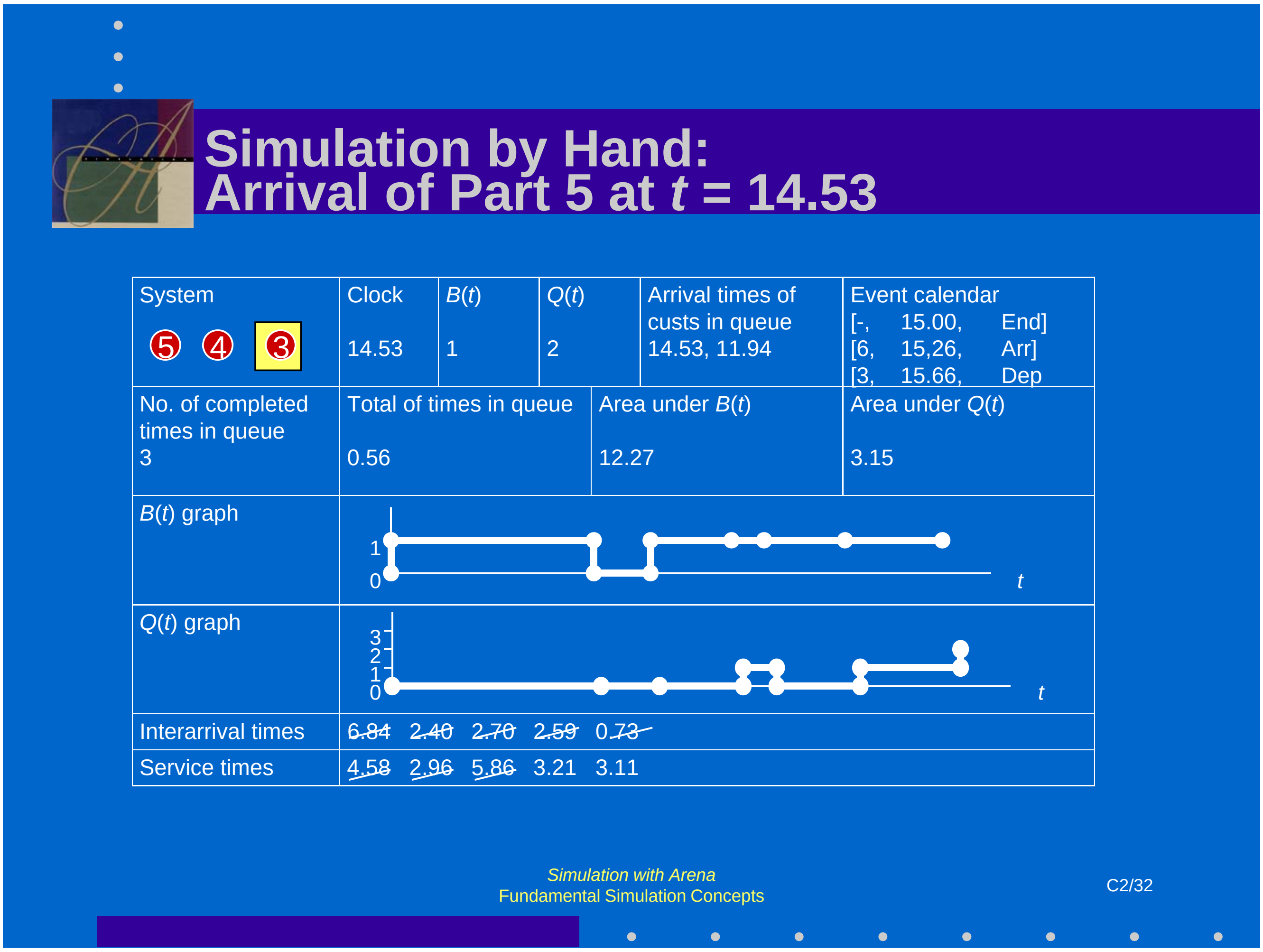
System	Clock	*B*(*t*)	*Q*(*t*)	Arrival times of custs in queue	Event calendar
4 3	11.94	1	1	11.94	[5, 14.53, Arr] [-, 15.00, End] [3, 15.66, Dep]

No. of completed times in queue	Total of times in queue	Area under *B*(*t*)	Area under *Q*(*t*)
3	0.56	9.68	0.56

B(*t*) graph

Q(*t*) graph

Interarrival times	~~6.84~~ ~~2.40~~ ~~2.70~~ ~~2.59~~ 0.73
Service times	~~4.58~~ ~~2.96~~ ~~5.86~~ 3.21 3.11



Simulation by Hand: Arrival of Part 5 at *t* = 14.53

System	Clock	*B*(*t*)	*Q*(*t*)	Arrival times of custs in queue	Event calendar
5 4 3	14.53	1	2	14.53, 11.94	[-, 15.00, End] [6, 15,26, Arr] [3, 15.66, Dep]

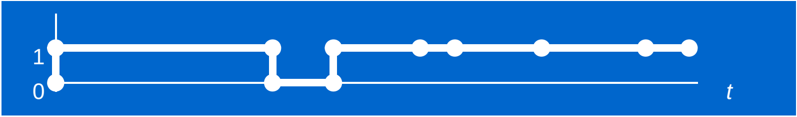
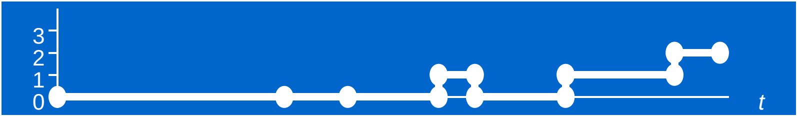
No. of completed times in queue	Total of times in queue	Area under *B*(*t*)	Area under *Q*(*t*)
3	0.56	12.27	3.15

B(*t*) graph

Q(*t*) graph

Interarrival times	~~6.84~~ ~~2.40~~ ~~2.70~~ ~~2.59~~ ~~0.73~~
Service times	~~4.58~~ ~~2.96~~ ~~5.86~~ 3.21 3.11

Simulation by Hand: The End at *t* = 15.00

<table>
<tr><td>System
5 4 3</td><td>Clock
15.00</td><td><i>B</i>(<i>t</i>)
1</td><td><i>Q</i>(<i>t</i>)
2</td><td colspan="2">Arrival times of custs in queue
14.53, 11.94</td><td colspan="2">Event calendar
[6, 15,26, Arr]
[3, 15.66, Dep]</td></tr>
<tr><td>No. of completed times in queue
3</td><td colspan="3">Total of times in queue
0.56</td><td colspan="2">Area under <i>B</i>(<i>t</i>)
12.74</td><td colspan="2">Area under <i>Q</i>(<i>t</i>)
4.09</td></tr>
<tr><td><i>B</i>(<i>t</i>) graph</td><td colspan="7"> </td></tr>
<tr><td><i>Q</i>(<i>t</i>) graph</td><td colspan="7"> </td></tr>
<tr><td>Interarrival times</td><td colspan="7"><s>6.84</s> <s>2.40</s> <s>2.70</s> <s>2.59</s> <s>0.73</s></td></tr>
<tr><td>Service times</td><td colspan="7"><s>4.58</s> <s>2.96</s> <s>5.86</s> 3.21 3.11</td></tr>
</table>

Simulation by Hand: Finishing Up

- **Average time in queue:**

$$\frac{\text{Total of times in queue}}{\text{No. of times in queue completed}} = \frac{0.56}{3} = 0.19 \text{ min./part}$$

- **Time-average number in queue:**

$$\frac{\text{Area under } Q(t) \text{ curve}}{\text{Final clock value}} = \frac{4.09}{15} = 0.27 \text{ part}$$

- **Server utilization**

$$\frac{\text{Area under } B(t) \text{ curve}}{\text{Final clock value}} = \frac{12.74}{15} = 0.85 \text{ (dimensionless)}$$

Complete Record of the Hand Simulation

Just-Finished Event			Variables		Attributes		Statistical Accumulators									Event Calendar
Entity No.	Time t	Event Type	$Q(t)$	$B(t)$	Arrival Times: (In Queue)	In Service	P	N	ΣD	D^*	ΣF	F^*	$\int Q$	Q^*	$\int B$	[Entity No., Time, Type]
–	0.00	Init	0	0	( )	___	0	0	0.00	0.00	0.00	0.00	0.00	0	0.00	[1, 0.00, Arr] [–, 15.00, End]
1	0.00	Arr	0	1	( )	0.00	0	1	0.00	0.00	0.00	0.00	0.00	0	0.00	[1, 4.58, Dep] [2, 6.84, Arr] [–, 15.00, End]
1	4.58	Dep	0	0	( )	___	1	1	0.00	0.00	4.58	4.58	0.00	0	4.58	[2, 6.84, Arr] [–, 15.00, End]
2	6.84	Arr	0	1	( )	6.84	1	2	0.00	0.00	4.58	4.58	0.00	0	4.58	[3, 9.24, Arr] [2, 9.80, Dep] [–, 15.00, End]
3	9.24	Arr	1	1	(9.24)	6.84	1	2	0.00	0.00	4.58	4.58	0.00	1	6.98	[2, 9.80, Dep] [4, 11.94, Arr] [–, 15.00, End]
2	9.80	Dep	0	1	( )	9.24	2	3	0.56	0.56	7.54	4.58	0.56	1	7.54	[4, 11.94, Arr] [–, 15.00, End] [3, 15.66, Dep]
4	11.94	Arr	1	1	(11.94)	9.24	2	3	0.56	0.56	7.54	4.58	0.56	1	9.68	[5, 14.53, Arr] [–, 15.00, End] [3, 15.66, Dep]
5	14.53	Arr	2	1	(14.53, 11.94)	9.24	2	3	0.56	0.56	7.54	4.58	3.15	2	12.27	[–, 15.00, End] [6, 15.26, Arr] [3, 15.66, Dep]
–	15.00	End	2	1	(14.53, 11.94)	9.24	2	3	0.56	0.56	7.54	4.58	4.09	2	12.74	[6, 15.26, Arr] [3, 15.66, Dep]

Event-Scheduling Logic via Programming

- **Clearly well suited to standard programming**
- **Often use “utility” libraries for:**
  - List processing
  - Random-number generation
  - Random-variate generation
  - Statistics collection
  - Event-list and clock management
  - Summary and output
- **Main program ties it together, executes events in order**

Simulation Dynamics: The Process-Interaction World View

- Identify characteristic *entities* in the system
- Multiple copies of entities co-exist, interact, compete
- “Code” is non-procedural
- Tell a “story” about what happens to a “typical” entity
- May have many types of entities, “fake” entities for things like machine breakdowns
- Usually requires special simulation software
  - Underneath, still executed as event-scheduling
- The view normally taken by Arena

Randomness in Simulation

- The above was just one “replication” -- a sample of size one (not worth much)
- Made a total of five replications:

Replication	1	2	3	4	5	Avg.	Std.Dev.
Avg. time in queue	0.19	1.12	3.72	0.00	0.00	1.01	1.59
Avg. no. in queue	0.27	0.30	0.99	0.00	0.00	0.31	0.41
Machine Utilization	0.85	0.93	1.00	0.32	0.37	0.69	0.32

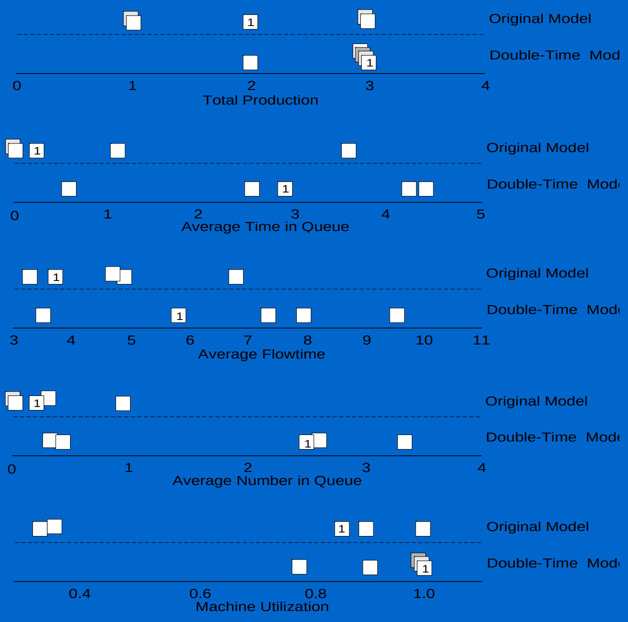
- Confidence intervals on expectations
  - e.g., 95% c.i. on E(avg. time in queue):

$$101 \pm 2.776 \frac{159}{\sqrt{5}}, \text{ or } 101 \pm 197$$

Comparing Alternatives

- Usually, simulation is used for more than just a single model "configuration"
- Often want to compare alternatives, select or search for the best (via some criterion)
- Simple processing system: What would happen if the arrival rate were to double?
  - Cut interarrival times in half
  - Rerun the model for double-time arrivals
  - Make five replications

Results:
Original vs. Double-Time Arrivals



- Note variability
- Danger of making decisions based on one (first) replication
- Hard to see if there are really differences
- Need: Statistical analysis of simulation output data

Overview of a Simulation Study

- Understand the system
- Be clear about the goals
- Formulate the model representation
- Translate into modeling software
- Verify “program”
- Validate model
- Design experiments
- Make runs
- Analyze, get insight, document results

Chapter 3
A Quick Peek at Arena

By : Arya Wirabhuana
Department Of Industrial Engineering
UIN SUNAN KALIJAGA YOGYAKARTA

What We'll Do ...

- **Start Arena**
- **Load, explore, run an existing model**
  - Basically the same model as for the hand simulation in Chapter 2
- **Arena is a true Windows 95/NT application**
  - Appearance is standard
  - Operation, functions are all there and standard
  - Interoperability with other software, e.g., MS Office
  - Proper memory management

Starting Up

- **Locate icon or shortcut; double-click**
  - Hardware key vs. Educational, Evaluation mode
- **Welcome window**
  - Tip of the Day (Next Tip, one way to Help system)
  - Alternate tabs at bottom for:
    - Introduction—Quick Preview, Jump-Start Wizard, SMARTs Library, Example Models, vendor contact info
    - About—version number
  - Future Tips? (Expert? None?)
  - Close button
- **Quitting Arena: File/Exit or Alt+F4 or top right button ☒**

Opening an Existing Model

- **File/Open … or 📂 button**
  - Navigate to desired disk/directory
  - Click/Open or double-click **`Mod_03_1.doe`**
- **Model window**
  - Where model is built
  - Resize, maximize, minimize, scroll/pan, zoom
  - Can have multiple model windows open at once
- **Template toolbar at left**
  - Tabs
  - Alternate modeling levels
  - Attach other panels

*Why the **`.doe`** default extension to Arena model files?*

Viewing the Model Window

- Underlying *world space* for model
- *Pan* with scroll bars or arrow keys
- *Zoom* in (down) with ⊞ button or + key
- Zoom out (up) with ⊟ button or - key
- See all at min altitude: 🔍 button or * key
- *Named views*
  - Save a pan/zoom view for different parts of model
  - Assign a *Hot key* (case-sensitive)
  - Access via View/Named Views … or ? key
- Display *grid* (▦), *snap* to grid (▦) toggles

Modules

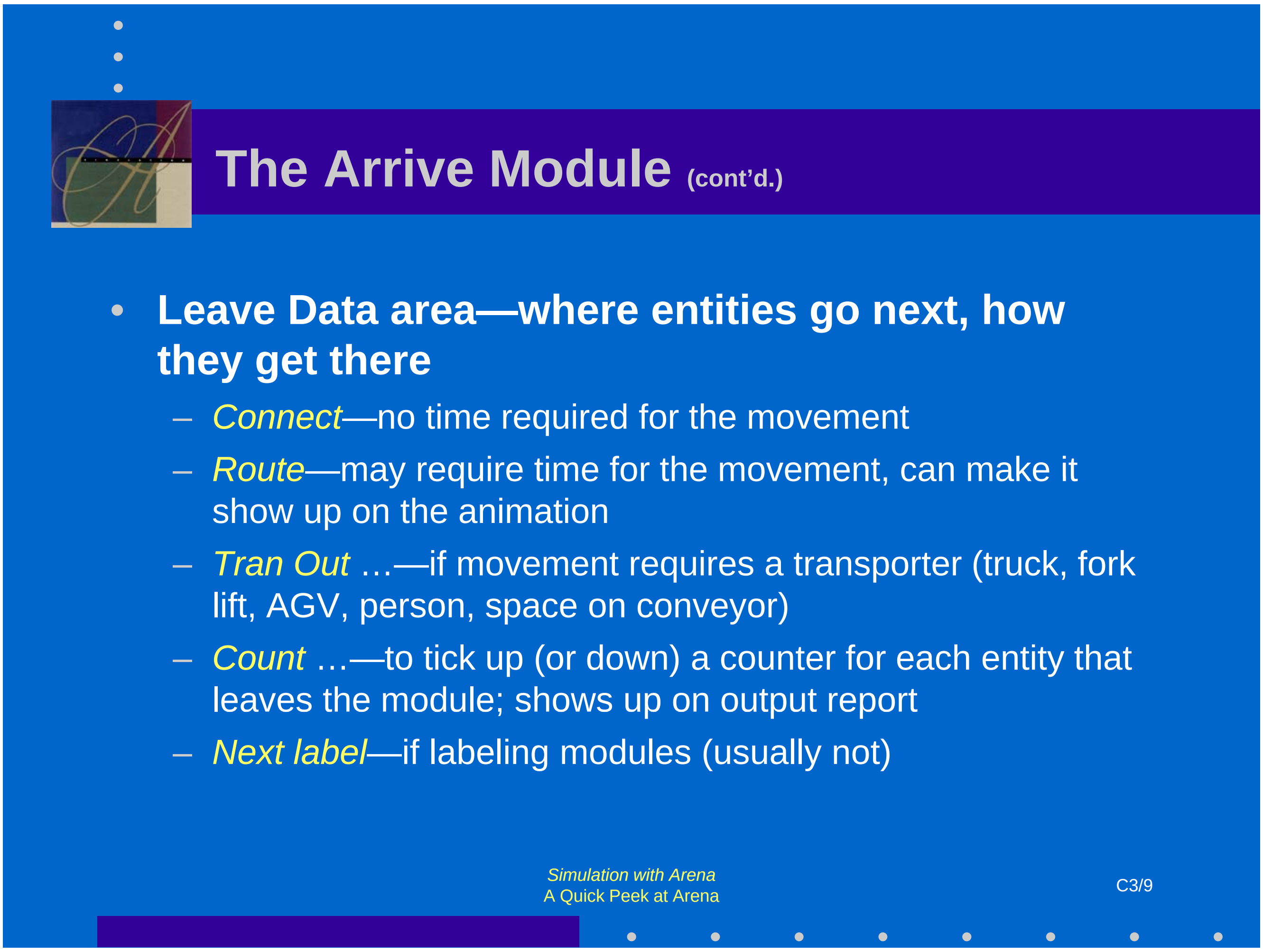
- Represent the “pieces” of a simulation model
- Two basic types:
  - *Logic modules*—entities flow through, are connected or routed to each other
  - *Data modules*—no entity flow, no connections, pertain to entire model
- Open an existing module: double-click on it
  - Main dialog, several levels of subdialogs
- Place a new module
  - (Single-) click from Template toolbar
  - Crosshairs, drop (click) where desired

The Arrive Module

- **“Birth” node for entities**
- **Double-click on module *handle*—module name inside the outline box**
- **Enter Data *area* — generally, how entities “enter” a logic module**
  - Station name
    - Required since it’s dark gray (has a default)
    - Physical location for animation

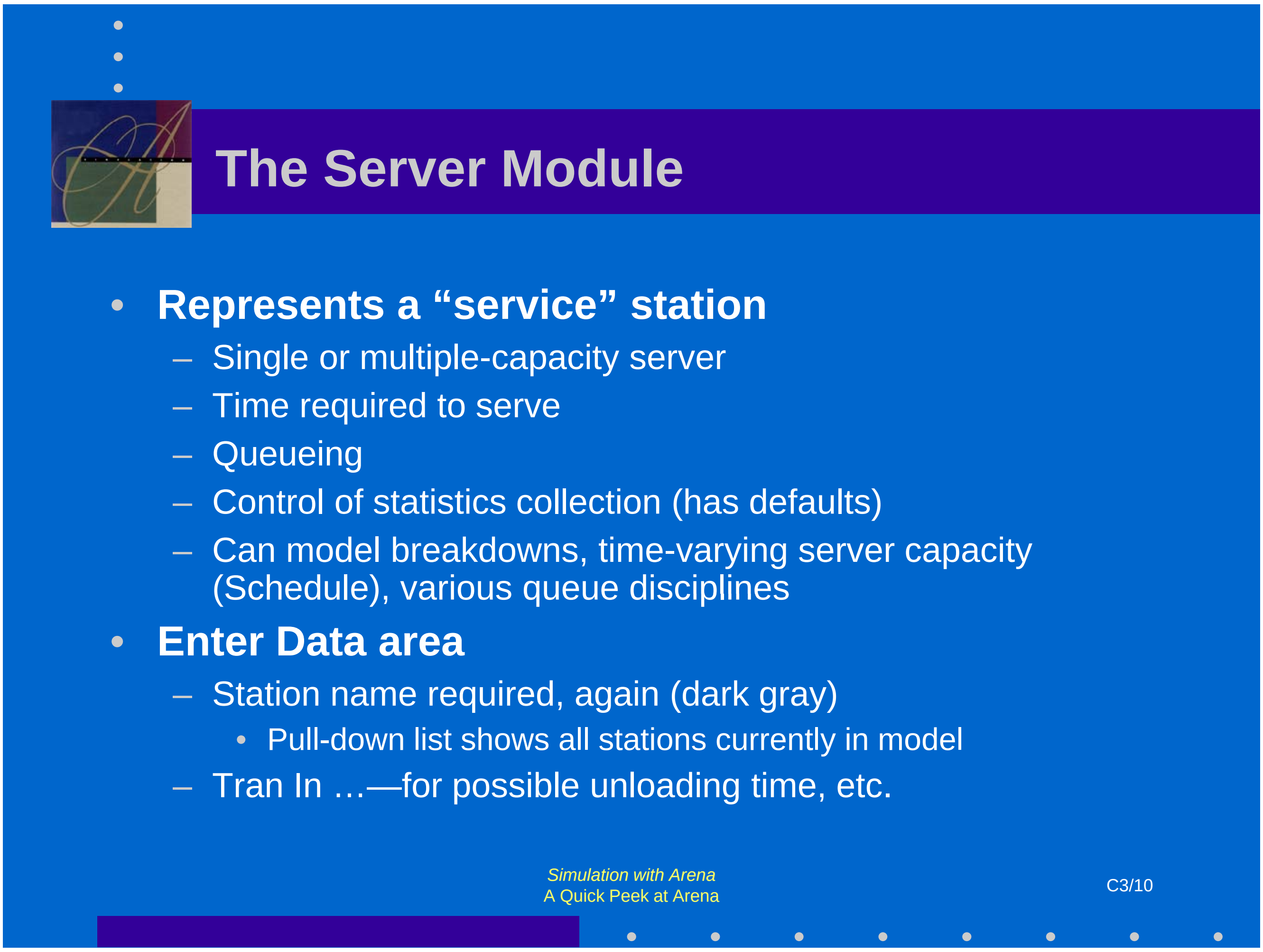
The Arrive Module (cont’d.)

- **Arrival Data area—what happens “during” this action**
  - *Batch Size* (default = 1)
  - Time of *First Creation* (default = 0)
  - *Time Between* successive creations
    - Often a distribution—but could be anything
    - Pull-down menu (▼) for distribution names, parameters
  - *Max* number of *Batches* to create—Arrive module will then “dry up”
  - *Mark* arrival Time Attribute with current clock time
  - *Assign* other attributes/variables, change animation



The Arrive Module (cont'd.)

- **Leave Data area—where entities go next, how they get there**
  - *Connect*—no time required for the movement
  - *Route*—may require time for the movement, can make it show up on the animation
  - *Tran Out* …—if movement requires a transporter (truck, fork lift, AGV, person, space on conveyor)
  - *Count* …—to tick up (or down) a counter for each entity that leaves the module; shows up on output report
  - *Next label*—if labeling modules (usually not)



The Server Module

- **Represents a "service" station**
  - Single or multiple-capacity server
  - Time required to serve
  - Queueing
  - Control of statistics collection (has defaults)
  - Can model breakdowns, time-varying server capacity (Schedule), various queue disciplines
- **Enter Data area**
  - Station name required, again (dark gray)
    - Pull-down list shows all stations currently in model
  - Tran In …—for possible unloading time, etc.

The Server Module (cont'd.)

- **Server Data area**
  - Name of *Resource* (required)
  - Fixed *Capacity* vs. time-varying *Schedule*
  - Check box for collecting Resource Statistics
  - *Process Time*
  - Subdialogs for Resource (e.g., breakdowns), Queue (e.g., other disciplines), Animate (e.g., change entity picture after processing), Options
- **Leave Data area—similar to Arrive module**

The Depart Module

- **Represents entities leaving the system**
- **Enter Data area**
  - Required Station name
  - Other features as in Arrive, Server
- ***Count* area**
  - Can specify a counter to be ticked up (or down) as each entity departs
  - To count production, number of failed parts, etc.
- ***Tally* area**
  - To "tally" observational data, e.g. flowtimes
  - *Interval* (of time) on a previously Marked Attribute:

    (Current clock time) - (Value of named Attribute)

The Simulate Module

- **Documents the model, controls run conditions**
- **This is a data module—not connected to any other modules**
- ***Project* area**
  - Title, Analyst name, Date—there are defaults
- ***Replicate* area**
  - *Number* of replications
  - Clock value at *Beginning Time*
  - *Length of Replication* —one of the ways to stop
  - Options on *initializing* between replications
  - Option for *Warmup* period (clear statistics)

Module Connections

- **If Connect selected in Leave Data area of a module, must say to where**
  - *Visual connection*
    - Click button from Arrange toolbar (or Module/Connect)
    - Connect output doo-dads to input doo-dads
  - *Auto-connect* toggle for newly placed modules
  - Can give the *Label* of next module in dialog
    - Connection doesn’t show up on picture
    - “Leftover” feature from SIMAN simulation language
    - Useful if model is highly complex

Module Connections (cont’d.)

- **If Route selected in Leave Data, animate movement with button from Animate toolbar**
- **If Tran Out selected in Leave Data, animate movement depending on type of transport (Paths for Transporters, Conveyors)**

Dynamic Plots

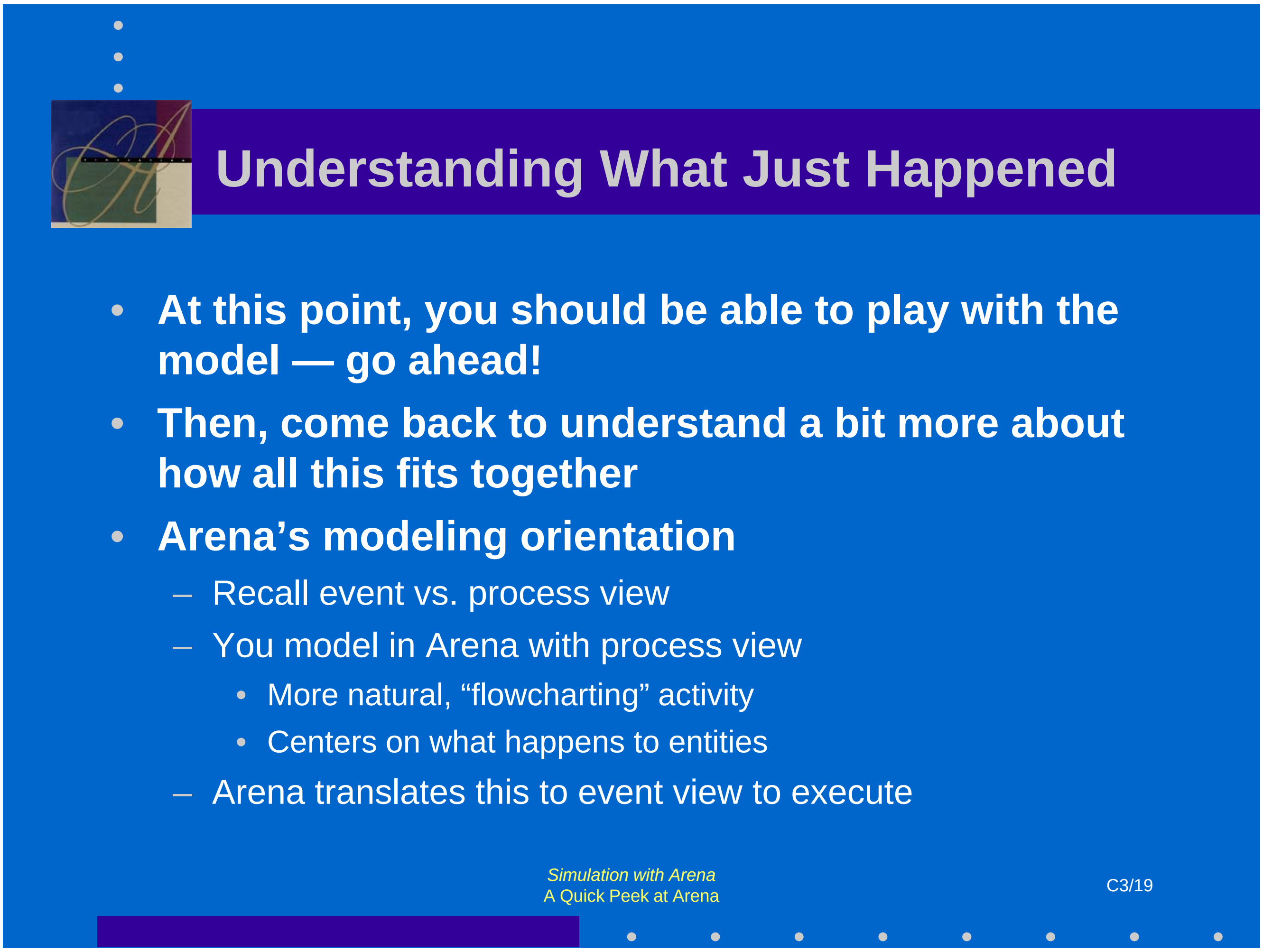
- **Trace variables, queues as simulation runs**
- **Disappear after run ends (to keep, must save data, postprocess via Output Analyzer — later)**
- **A kind of “data animation”**
- **Can plot via Plot button from Animate toolbar**
- **Easier: *Animate module* from Common panel**
  - Select “Data Object” to animate
  - Select “Information” about the object
  - Choose method(s) of Display
- **Configure the graph visually**

Dressing Things Up

- **Add drawing objects from Draw toolbar**
  - Similar to other drawing, CAD packages
  - Object-oriented drawing tools (layers, etc.), not just a painting tool
- **Add Text to annotate things**
  - Control font, size, color, orientation

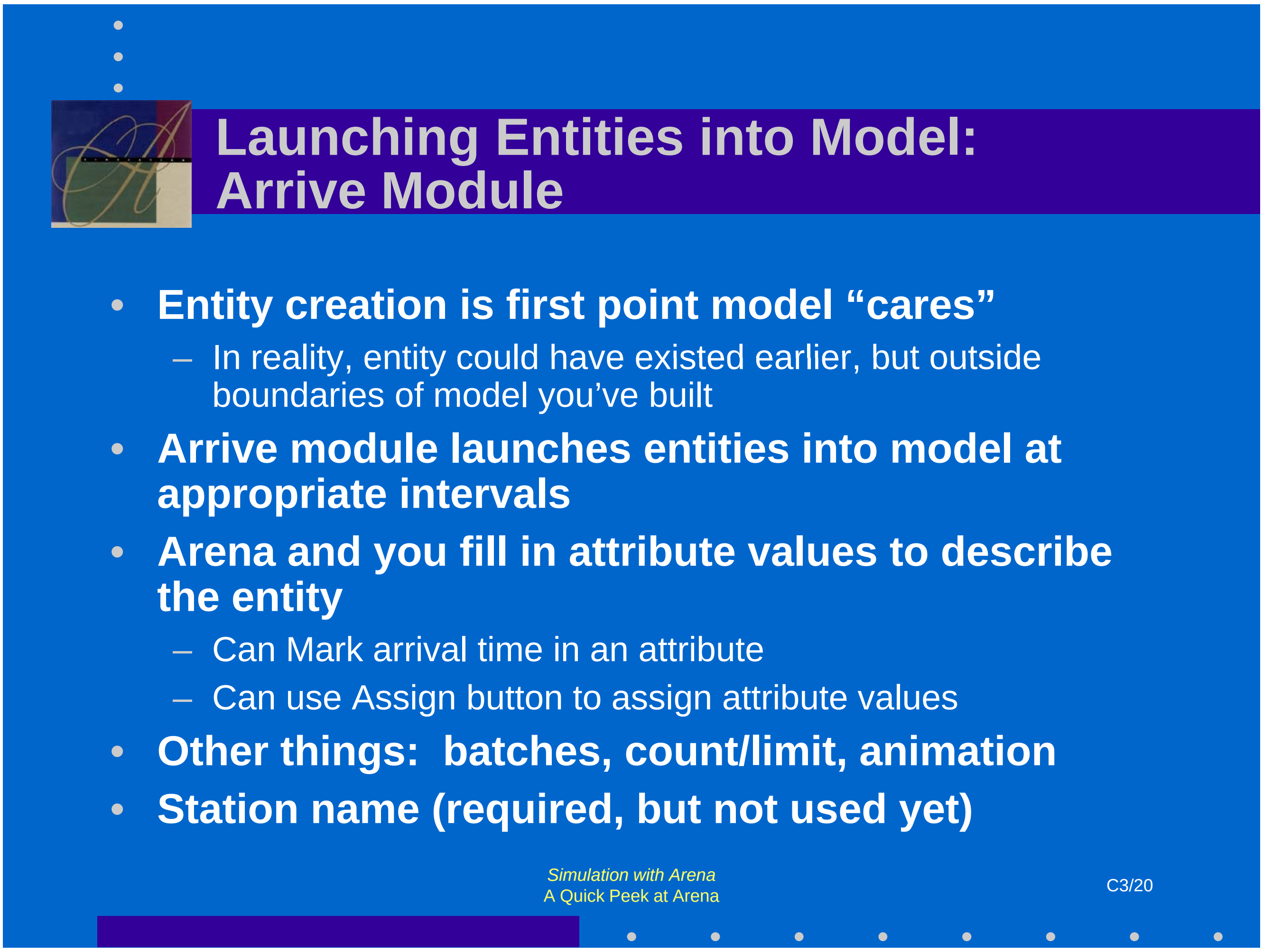
Running It

- **Plain-vanilla run: Click ▶ button from Run toolbar (like VCRs, CD players, etc.)**
  - First time or after changes: *Check*
  - Enters *run mode*—can move around but not edit
  - Speed up (>) or slow down (<) animation display
  - When done, asked if you want to see summary results (via MS Notepad or other text editor)
  - Click ■ button to get out of run mode
  - Can *pause* run with ⏸ button
- **Other run control, viewing, checking options**



Understanding What Just Happened

- At this point, you should be able to play with the model — go ahead!
- Then, come back to understand a bit more about how all this fits together
- Arena’s modeling orientation
  - Recall event vs. process view
  - You model in Arena with process view
    - More natural, “flowcharting” activity
    - Centers on what happens to entities
  - Arena translates this to event view to execute



Launching Entities into Model: Arrive Module

- Entity creation is first point model “cares”
  - In reality, entity could have existed earlier, but outside boundaries of model you’ve built
- Arrive module launches entities into model at appropriate intervals
- Arena and you fill in attribute values to describe the entity
  - Can Mark arrival time in an attribute
  - Can use Assign button to assign attribute values
- Other things: batches, count/limit, animation
- Station name (required, but not used yet)

Processing the Entity: Server Module

- **Represents processing**
  - Resource
    - Name (defaults to StationName_R)
    - How many units?
    - Fixed Capacity of units or Schedule number during run?
    - Downtimes, Failures?
    - Status, statistics: Busy, Idle, Failed, … others
    - Animation picture in different states
  - Queue to wait for resource (if necessary)
    - FIFO, LIFO, or ranked?
  - Service process (time required)
- **Really three activities combined; can break up**

Leaving the Model: Depart Module

- **After going through Depart, entity ceases to exist in model (though in reality could go on—model boundary)**
- **Opportunity to collect statistics**
  - Count (by ones or anything else)
  - Tally interval of time, other things
- **Required Station name**

Controlling the Run: Simulate Module

- **Data module, not logic module**
- **Type of run to do**
  - Limit length of simulated time for a replication
    - One way to “turn off” a run
    - There are other ways to stop
    - Can have multiple stopping rules—use first one to “hit”
  - Number of replications
  - “Warm Up” model before retaining statistics
- **Default entity animation picture (red box)**
- **Need only a single Simulate module**

Chapter 4 . Working with Arena

By : Arya Wirabhuana
Department Of Industrial Engineering
UIN SUNAN KALIJAGA YOGYAKARTA

What We'll Do ...

- User interface
- Menus (including Running)
- Toolbars
- Help
- Model windows
- Drawing
- Printing
- Running
- Building the simple processing model

Basic Interaction

- **True Windows 95/NT application**
  - Usual behavior of mouse, keyboard, selecting, dragging, resizing, menus, buttons, folder tabs
  - Usual window operations — resize, reposition, layer, maximize, minimize, close
- **Usual file operations**
  - New, Open, Save, Save As, Print, Print Preview
- **Cut/Copy/Paste**
  - Within an Arena window
  - Between Arena windows
  - Between Arena, other apps (when sensible)

Basic Interaction (cont'd.)

- **Miscellaneous**
  - Object-or select an object, then act on it
  - Undo/Redo
  - Right (secondary) mouse button
    - Repeat last action
    - Bail out of current action
  - Drawing: hold Shift to constrain lines to 45^0, rectangles to squares, ellipses to circles, etc.
  - Cycle among open Arena windows: Ctrl+Tab
  - Duplicate selection: Ctrl+D — then move it, edit it

Menus

- **File, Edit, View, Tools, Arrange, Module, Run, Window, Help**
  - If no model window open: just File, View, Tools, Help
- **Standard Windows operations in usual menus**
  - File menu has Open, Save, Save As, etc.
  - Edit menu has Cut, Copy, Paste, etc.
  - Help menu
  - etc.
  - So focus below on Arena-specific things in menus
- **Many menu items have toolbar buttons too**

File Menu

- **New, Open..., Save, Save As...**
- **Import CAD DXF files for "backdrop"**
- **Color palette**
- **Print, Print Preview**
- **Send mail**
  - Attach .doe files
- **Recent models**
- **Exit Arena**

Edit Menu

- Undo/Redo
- Cut, Copy, Paste
- Paste Link (OLE link)
- Duplicate, Delete selected object
- Select All, Deselect All
- Find — searches through all modules in current model for text string (for slobs who forget what names they’ve chosen, or for poor typists)
- Links — info, control links to other kinds of documents
- Insert, edit Objects from other applications

View Menu

- Zooming In (down), Out (up)
- Named views
  - Pre-defined by Arena, you
  - Go to defined views (hot keys better)
- Grid — display, Snap, settings
- Layers — what model objects show u hat mode
- Toolbars — what “button” collections show up
- Status b (check, uncheck) — state, coords.

Tools Menu

- Separate applications related to Arena model building and analysis
  - Input Analyzer
  - Output Analyzer
- Scenario Manager
- License information
- Add-Ins installed
- Visual Basic Editor to write VBA code for model (see Chapter 10)
- Options… — control many aspects of how Arena works, looks

Arrange Menu

- For drawing objects
- Bring selected object to Front, Send it to Back — for “stacking” effects
- Group, Ungroup objects
- Flip object around Vertical, Horizontal line
- Rotate object
- Snap selected object to Grid
- Change point on object that gets snapped

Module Menu

- **Connection ntrols**
  - Connect tool
  - Auto-Connect newly placed module to selected module — toggle on/off
  - Smart Connections — new connections in horizontal/vertical segments rather than one diagonal segment — toggle on/off

Module Menu (cont'd.)

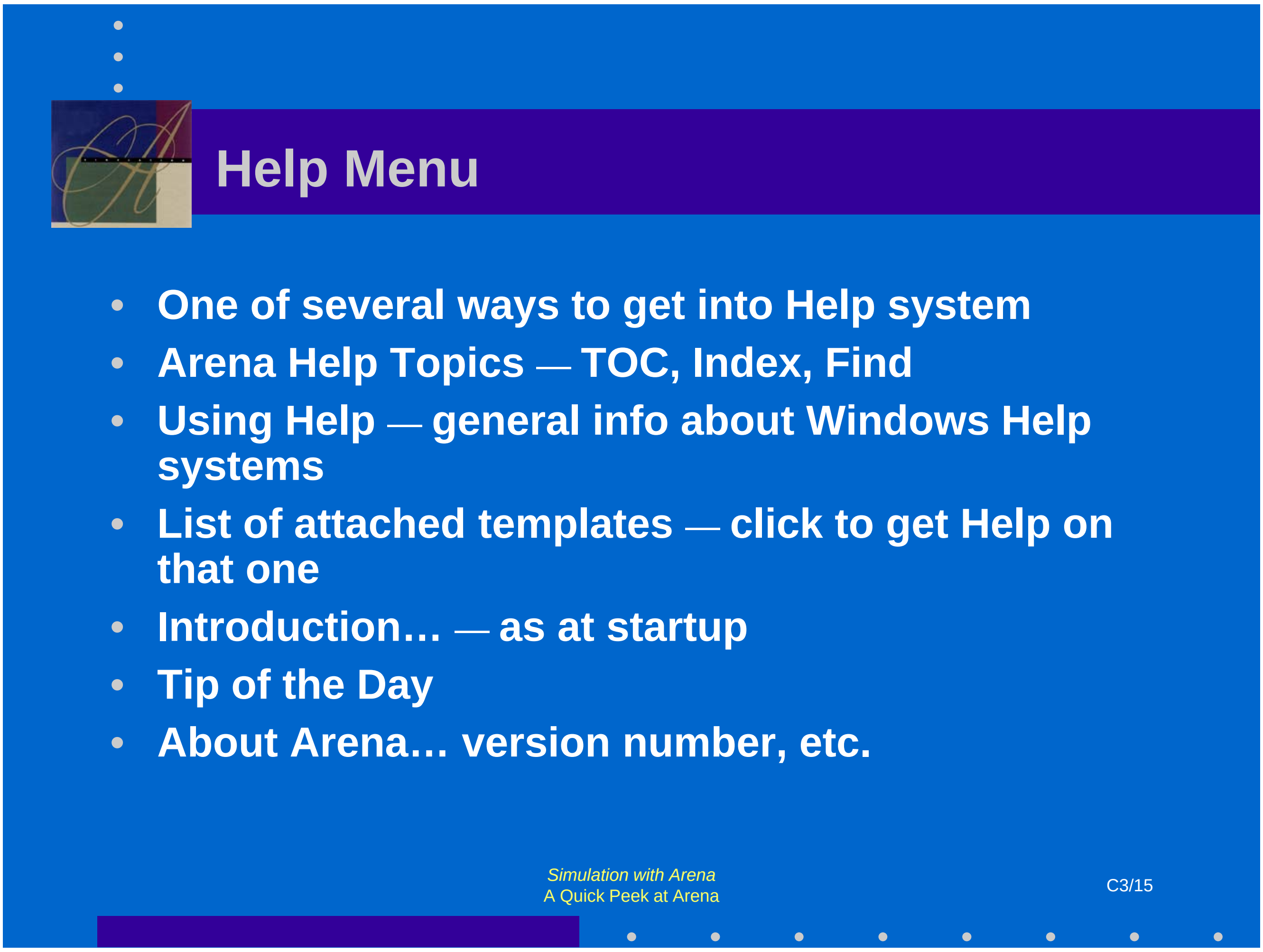
- **Template Panel**
  - Attach (easier to use Attach… tab)
  - Detach — even if model has modules from it
- **Show — toggle for module's name on screen or just its animation picture**
- **Breakpoint — attach to module to interrupt during run (for debugging)**

Run Menu

- Entries to run, check, pause, step through
- Alternatives to watch execution, view results (or errors)
- Control how run goes and is displayed
- Most capabilities on Run or Run Interaction Toolbar, and will be described a bit later in detail

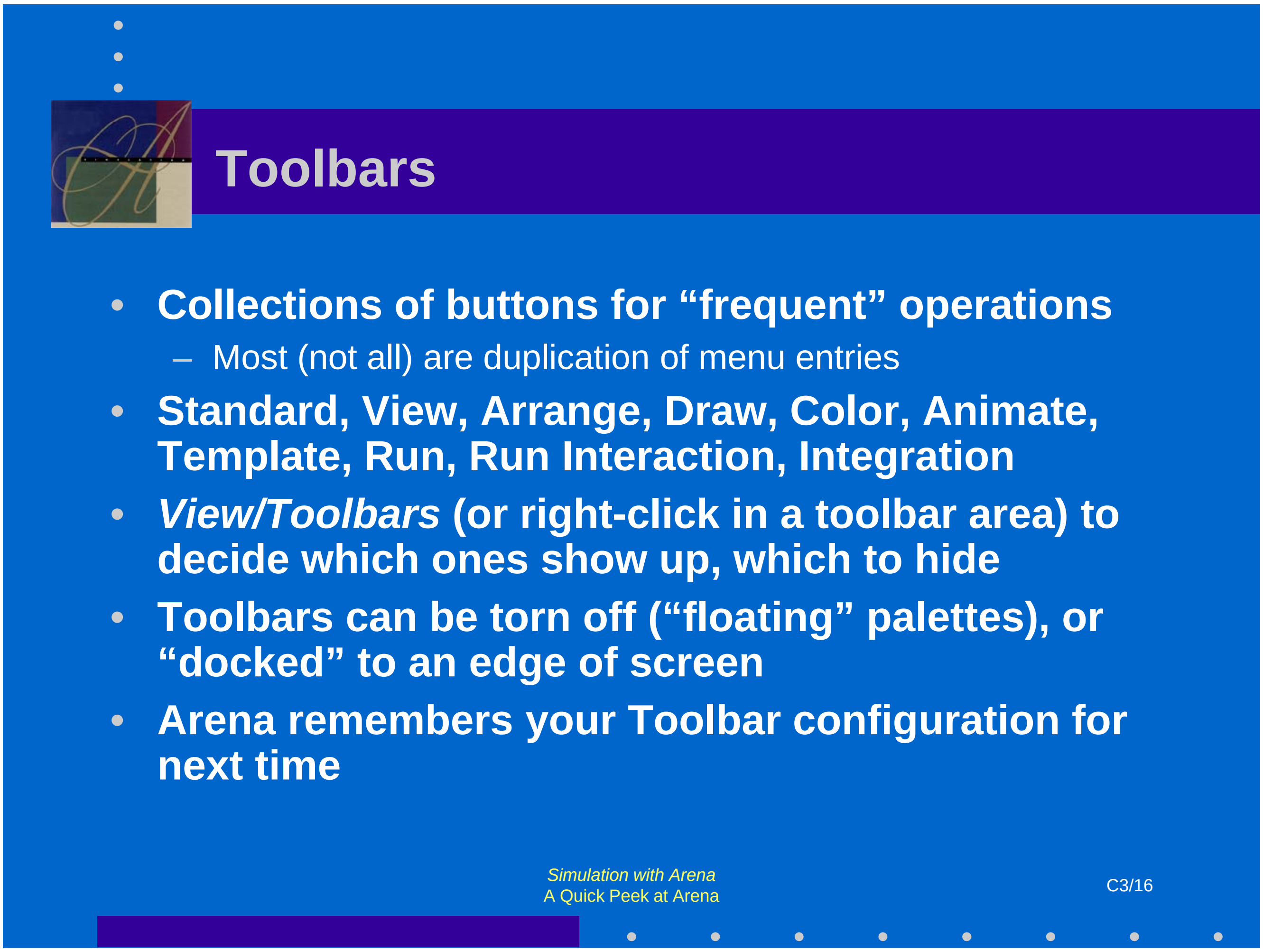
Window Menu

- Cascade, Tile multiple open model windows
- Arrange Icons for any minimized model windows
- Use system Background Color — use Windows colors rather than Arena settings
- List of open model windows



Help Menu

- One of several ways to get into Help system
- Arena Help Topics — TOC, Index, Find
- Using Help — general info about Windows Help systems
- List of attached templates — click to get Help on that one
- Introduction… — as at startup
- Tip of the Day
- About Arena… version number, etc.



Toolbars

- Collections of buttons for “frequent” operations
  - Most (not all) are duplication of menu entries
- Standard, View, Arrange, Draw, Color, Animate, Template, Run, Run Interaction, Integration
- *View/Toolbars* (or right-click in a toolbar area) to decide which ones show up, which to hide
- Toolbars can be torn off (“floating” palettes), or “docked” to an edge of screen
- Arena remembers your Toolbar configuration for next time

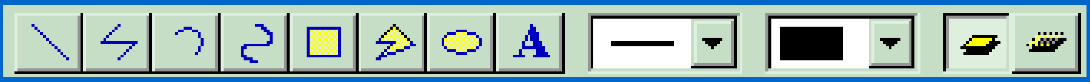
Help

- Extensive, comprehensive online system — replaces traditional manuals
- Interlinked via hypertext for cross referencing
- Multiple entry points, including Help menu
- Welcome screen described in Chapt. 3
- ↖? button for context-sensitive help
- Help button in many dialogs
- ? button (What’s This?) for info on items
- Tool tips — leave mouse motionless on something, get little yellow sticky note

Model Windows

- Where you build, edit, run a model
- Can have several open at once
- Really a “window” on the “world” — (*x*, *y*) coordinates in *world units*
- Open existing models (default .doe extension)
- Create new (blank) model windows
- Save, Save As...
- Add modeling constructs — click from template, drop into model

Drawing



- Line, Polyline (hold Shift key for 45^0), Arc, Bezier Curve
- Box, Polygon, Ellipse (fill, line, shade)
- Text (font, size, style, color)
- Visible/Hidden setting — show during animation?
- Change position, size, color
- By far best way to learn: just play around

Printing

- Can print directly from Arena
- Supports color
- Big models — many pages
- Also will print any Named Views separately
  - Print Preview, select only what you want for printing
- PrintScreen key — sends screen to clipboard, paste into another application
  - Alt+PrintScreen — sends only active window to clipboard

Running

- Run Menu; Run & Run Interaction Toolbars
- *Go* (▶) — run simulation “normally” (depends on Options... selected from Tools menu)
- *Step* (▶❚) — one “step” at a time (verify, debug)
- *Fast-Forward* (⏩) — disable animation (faster)
- *Pause* (❚❚) — freeze run, resume with Go
- *Start Over* (⏮) — go back to beginning of simulation
- *End* (■) — get out of run mode

Running (cont’d.)

- *Check Model* (✓) — like compiling
- *Review Errors* — for most recent Check
- *View Results* — Summary for most recent run
- SIMAN — see, save underlying SIMAN `.mod`, `.exp` files generated from your Arena model
- *Command* (▣) — bring up interactive command-line window to control run
- *Break*… (✋) — set times, conditions to interrupt for checks, illustration

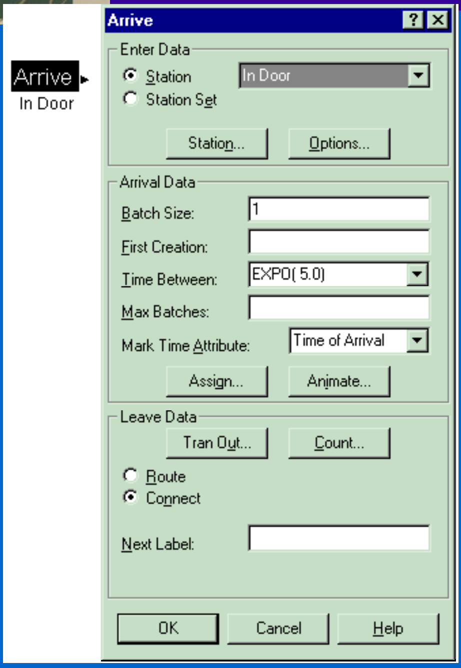
Running (cont'd.)

- *Trace*… (🗐) — generate detailed history (debugging)
- *Watch*… (👓) — bring up a window to watch a variable or expression during run
- *Report*… (▤) — see a numerical status when Paused
- *Setup*… — alter run conditions (animation on/off, full-screen during run animation)
- *Speed* — control speed of animation display (< and > keys easier)

Building the Simple Processing Model

- Open a new model window
- Attach Common template (if not auto-attached — check *Tools/Options/Toolbars* tab)
- Drop in Arrive, Server, Depart, Simulate, two Animate modules
- Double-click, edit modules as indicated in text
- Connect logic modules as required
- Dress up with drawing, text
- Run
- The "Display" notation used from here on
  - Module handle/animation, completed dialog(s), table showing what to enter (omitting defaults)

Display for the Arrive Module



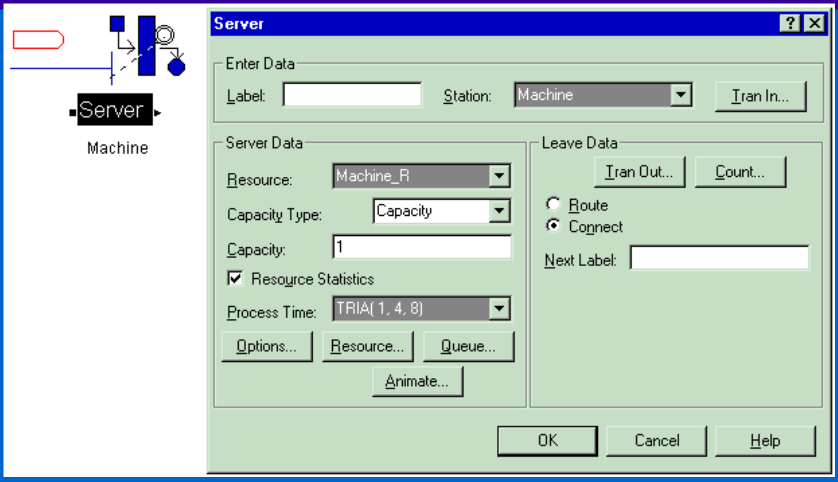
Enter Data
● Station
○ Station Set
In Door

Arrival Data
Batch Size: 1
First Creation:
Time Between: EXPO(5.0)
Max Batches:
Mark Time Attribute: Time of Arrival

Leave Data
○ Route
● Connect
Next Label:

Enter Data	
Station	`In Door`
Arrival Data	
Time Between	`EXPO(5.0)`
Mark Time Attribute	`Time of Arrival`
Leave Data	
Connect	*select*

Display for the Server Module



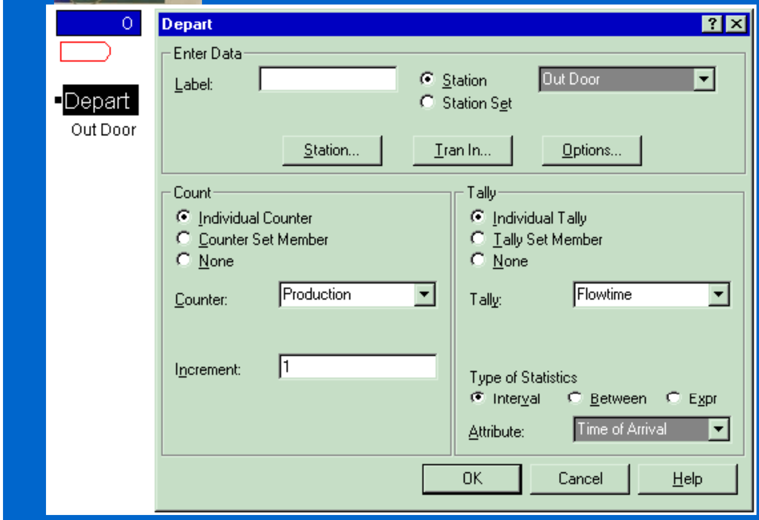
Enter Data
Label:
Station: Machine

Server Data
Resource: Machine_R
Capacity Type: Capacity
Capacity: 1
☑ Resource Statistics
Process Time: TRIA(1, 4, 8)

Leave Data
○ Route
● Connect
Next Label:

Enter Data	
Station	`Machine`
Server Data	
Process Time	`TRIA(1,4,8)`
Leave Data	
Connect	*select*

Display for the Depart Module



0

Depart
Out Door

Depart

Enter Data

Label:

● Station ○ Station Set

Out Door

Station... Tran In... Options...

Count

● Individual Counter
○ Counter Set Member
○ None

Counter: Production

Increment: 1

Tally

● Individual Tally
○ Tally Set Member
○ None

Tally: Flowtime

Type of Statistics
● Interval ○ Between ○ Expr

Attribute: Time of Arrival

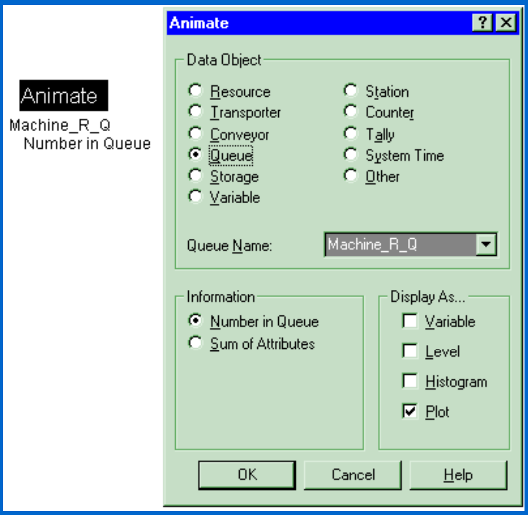
OK Cancel Help

Enter Data
  Station `Out Door`

Count
  Individual Counter *select*
  Counter `Productio`

Tally
  Individual Tally *select*
  Tally `Flowtime`
  Attribute `Time of Arrival`

Animate Module and Plot: Queue Length



Animate
Machine_R_Q
Number in Queue

Animate

Data Object

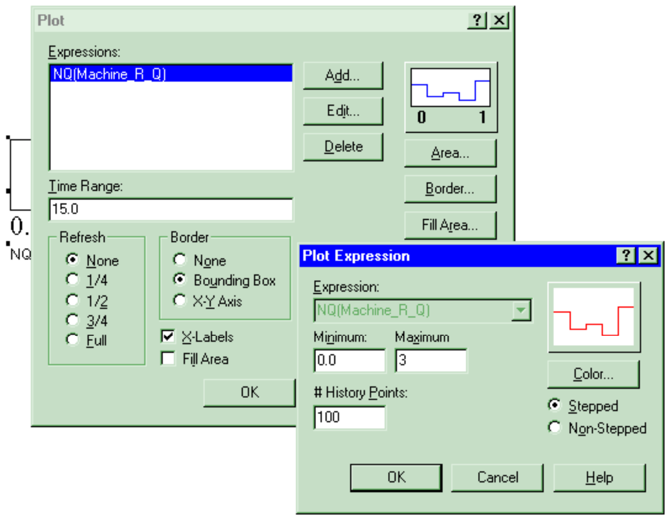
○ Resource ○ Station
○ Transporter ○ Counter
○ Conveyor ○ Tally
● Queue ○ System Time
○ Storage ○ Other
○ Variable

Queue Name: Machine_R_Q

Information
● Number in Queue
○ Sum of Attributes

Display As...
□ Variable
□ Level
□ Histogram
☑ Plot

OK Cancel Help



Plot

Expressions:
NQ(Machine_R_Q)

Add... Edit... Delete

Area... Border... Fill Area...

Time Range: 15.0

Refresh
● None ○ 1/4 ○ 1/2 ○ 3/4 ○ Full

Border
○ None ● Bounding Box ○ X-Y Axis

☑ X-Labels
□ Fill Area

OK

Plot Expression

Expression: NQ(Machine_R_Q)

Minimum: 0.0 Maximum: 3

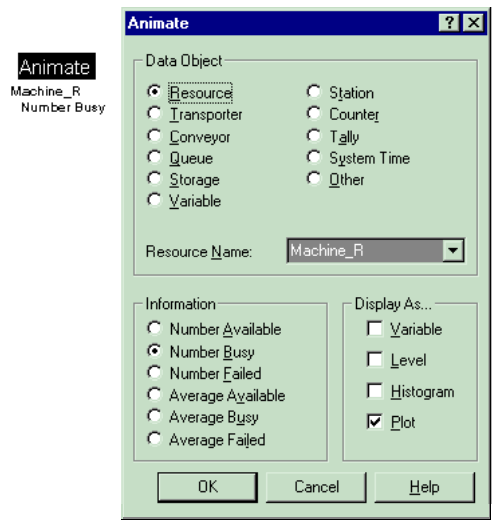
History Points: 100

Color...

● Stepped
○ Non-Stepped

OK Cancel Help

Animate Module and Plot: Number Busy



Animate
Machine_R
Number Busy

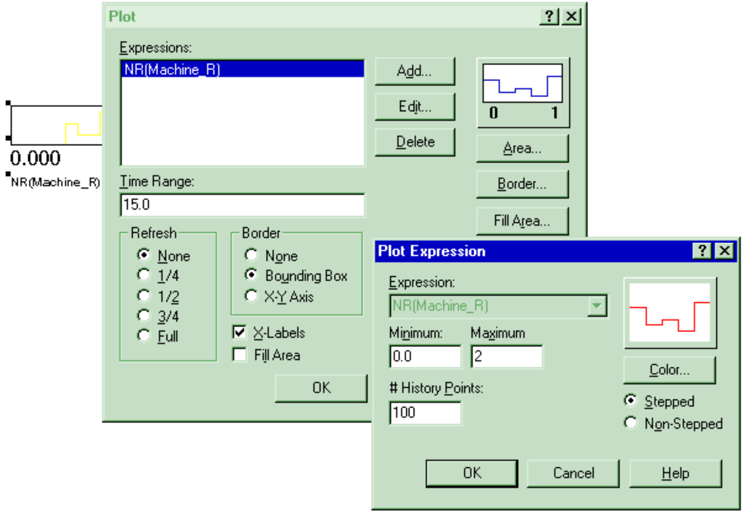
Animate

Data Object
- ● Resource
- ○ Transporter
- ○ Conveyor
- ○ Queue
- ○ Storage
- ○ Variable
- ○ Station
- ○ Counter
- ○ Tally
- ○ System Time
- ○ Other

Resource Name: Machine_R

Information
- ○ Number Available
- ● Number Busy
- ○ Number Failed
- ○ Average Available
- ○ Average Busy
- ○ Average Failed

Display As...
- ☐ Variable
- ☐ Level
- ☐ Histogram
- ☑ Plot



0.000
NR(Machine_R)

Plot

Expressions: NR(Machine_R)

Time Range: 15.0

Refresh
- ● None
- ○ 1/4
- ○ 1/2
- ○ 3/4
- ○ Full

Border
- ○ None
- ● Bounding Box
- ○ X-Y Axis

☑ X-Labels
☐ Fill Area

Plot Expression

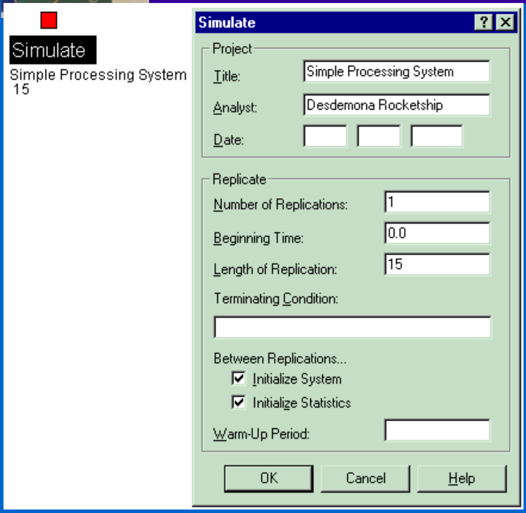
Expression: NR(Machine_R)

Minimum: 0.0
Maximum: 2

History Points: 100

- ● Stepped
- ○ Non-Stepped

Display for the Simulate Module



Simulate
Simple Processing System
15

Simulate

Project
Title: Simple Processing System
Analyst: Desdemona Rocketship
Date:

Replicate
Number of Replications: 1
Beginning Time: 0.0
Length of Replication: 15
Terminating Condition:

Between Replications...
☑ Initialize System
☑ Initialize Statistics

Warm-Up Period:

Project
Title Simple Processing System
Analyst Desdemona Rocketship

Replicate
Length of Replication 15

Chapter 5
Modeling Basic Operations and Inputs

By : Arya Wirabhuana

Department Of Industrial Engineering

UIN SUNAN KALIJAGA YOGYAKARTA

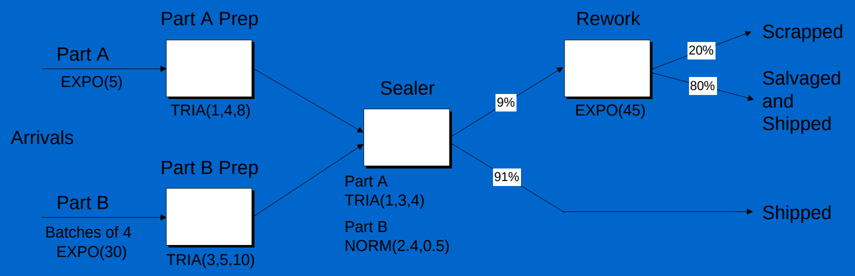
What We'll Do ...

- **Model 5.1: Electronic assembly/test system**
  - Modeling approaches
  - Stations, Transfers, Pictures
- **Model 5.2: Enhanced electronic assembly/test**
  - Resources, Resource States, Schedules
  - Saving statistical data
  - Output Analyzer (data display only)
- **Model 5.3: Enhancing the animation**
  - Queues, Entity Pictures, Resource Pictures
  - Adding Plots and Variables

What We'll Do ... (cont'd.)

- **Input analysis**
  - Specifying input distributions, parameters
  - Deterministic vs. random input
  - Collecting and using data
  - Fitting input distributions via the Input Analyzer
  - No data?
  - Nonstationary arrival processes
  - Multivariate and correlated input data

Electronic Assembly/Test System (Model 5.1)



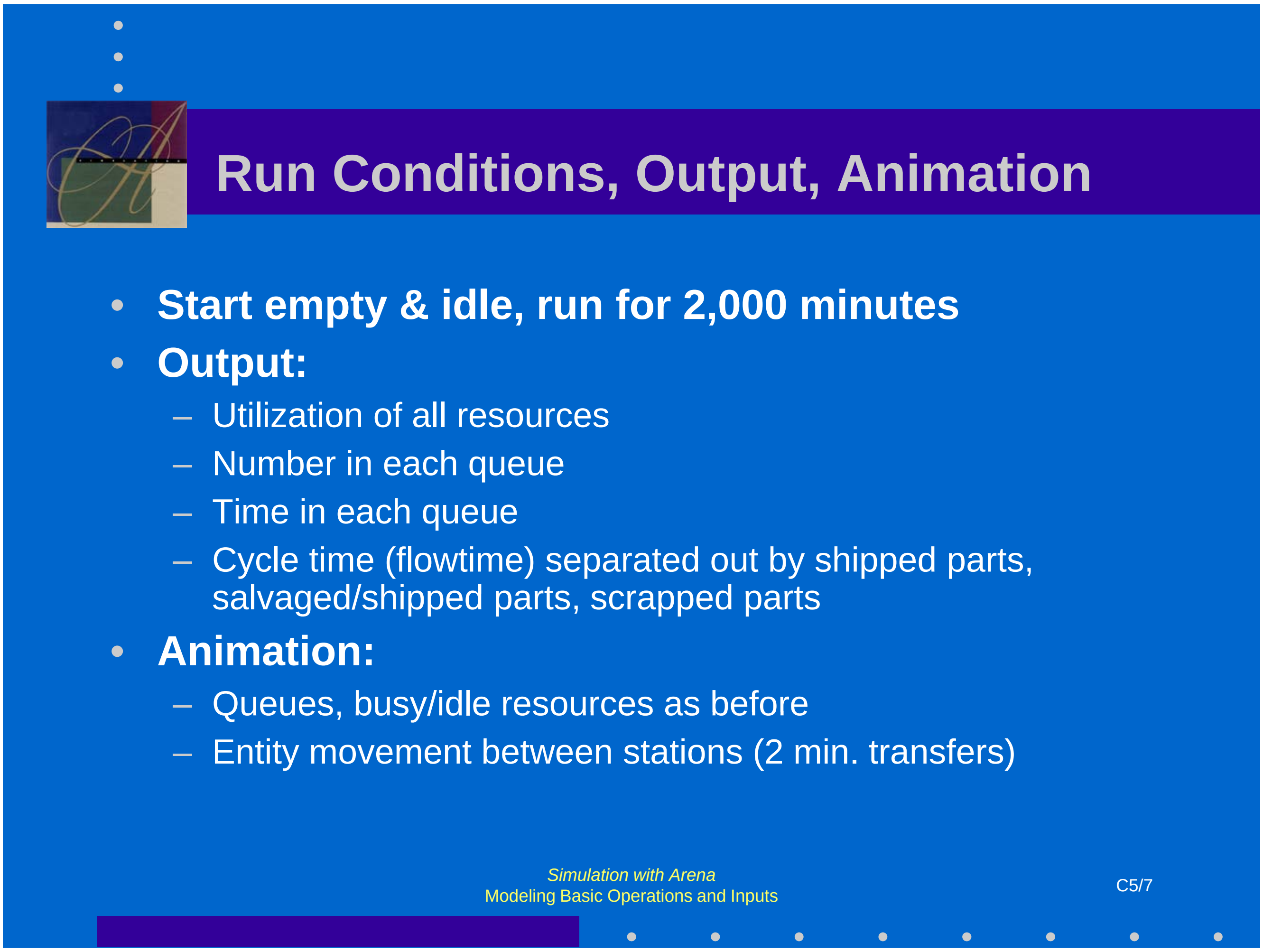
- **Produce two different sealed electronic units (A, B)**
- **Arriving parts: cast metal cases machined to accept the electronic parts**
- **Part A, Part B — separate prep areas**
- **Both go to Sealer for assembly, testing — then to Shipping (out) if OK, or else to Rework**
- **Rework — Salvage or Scrap**

Part A

- **Interarrivals: expo (5) minutes**
- **Transit times between all stations: 2 min.**
  – No wait for person, cart to transfer — parts have their own feet (relax this assumption in Chapt. 7)
- **Go to Part A Prep area**
  – Process = (machine + deburr + clean) ~ tria (1,4,8)
- **Go to Sealer**
  – Process = (assemble + test) ~ tria (1,3,4) min.
  – 91% pass, go to Shipped; Else go to Rework
- **Rework: (re-process + testing) ~ expo (45)**
  – 80% pass, go to Salvage/Ship; Else go to Scrap

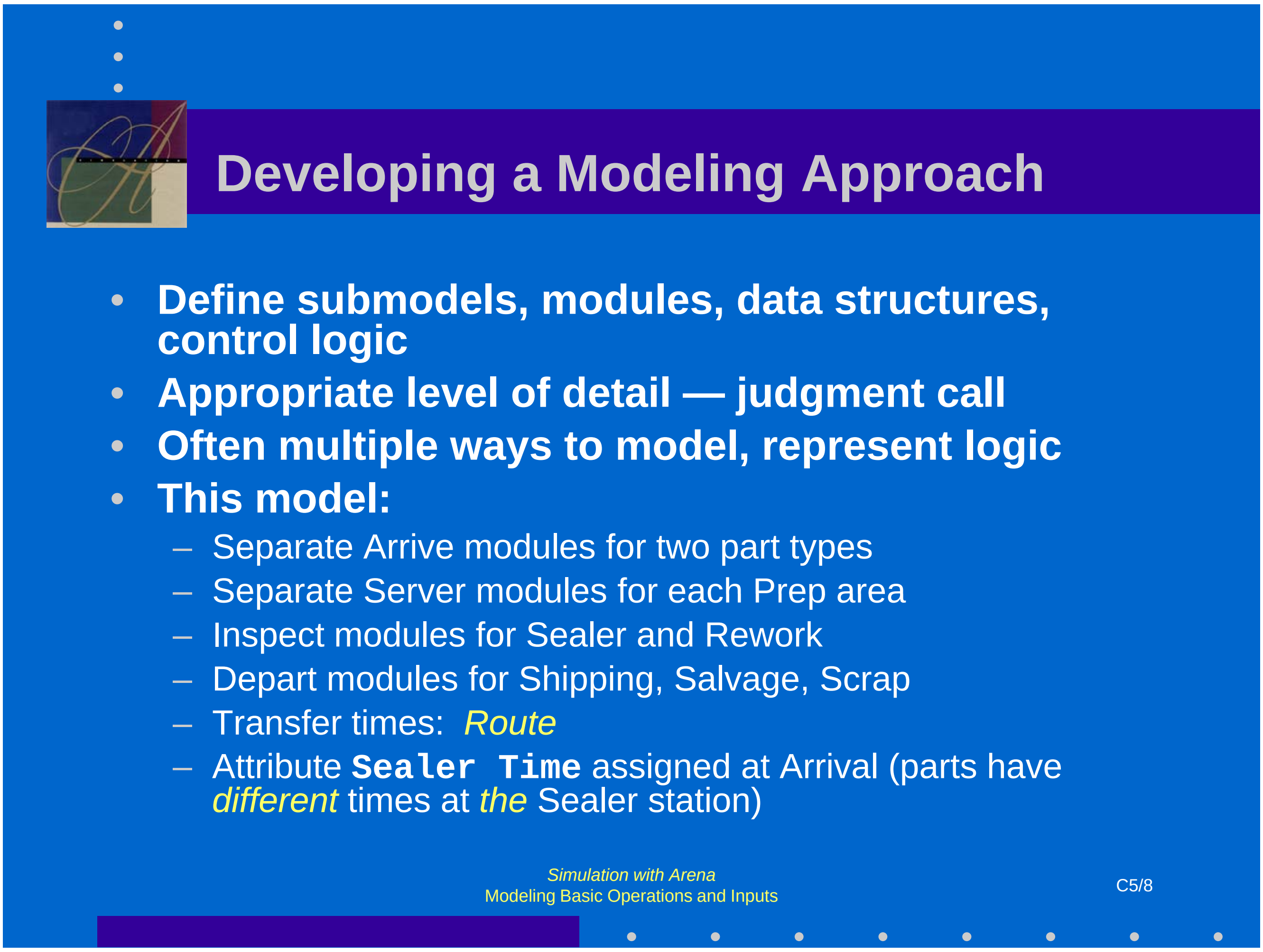
Part B

- **Interarrivals: *batches* of 4, expo (30) min.**
- **Transit times between all stations: 2 min.**
- **Go to Part B Prep area**
  – Process = (machine + deburr +clean) ~ tria (3,5,10)
- **Go to Sealer**
  – Process = (assemble + test) ~ norm (2.4, 0.5) min. , *different* from Part A, though at same station
  – 91% pass, go to Shipped; Else go to Rework
- **Rework: (re-process + test) = expo (45) min.**
  – 80% pass, go to Salvage/Ship; Else go to Scrap



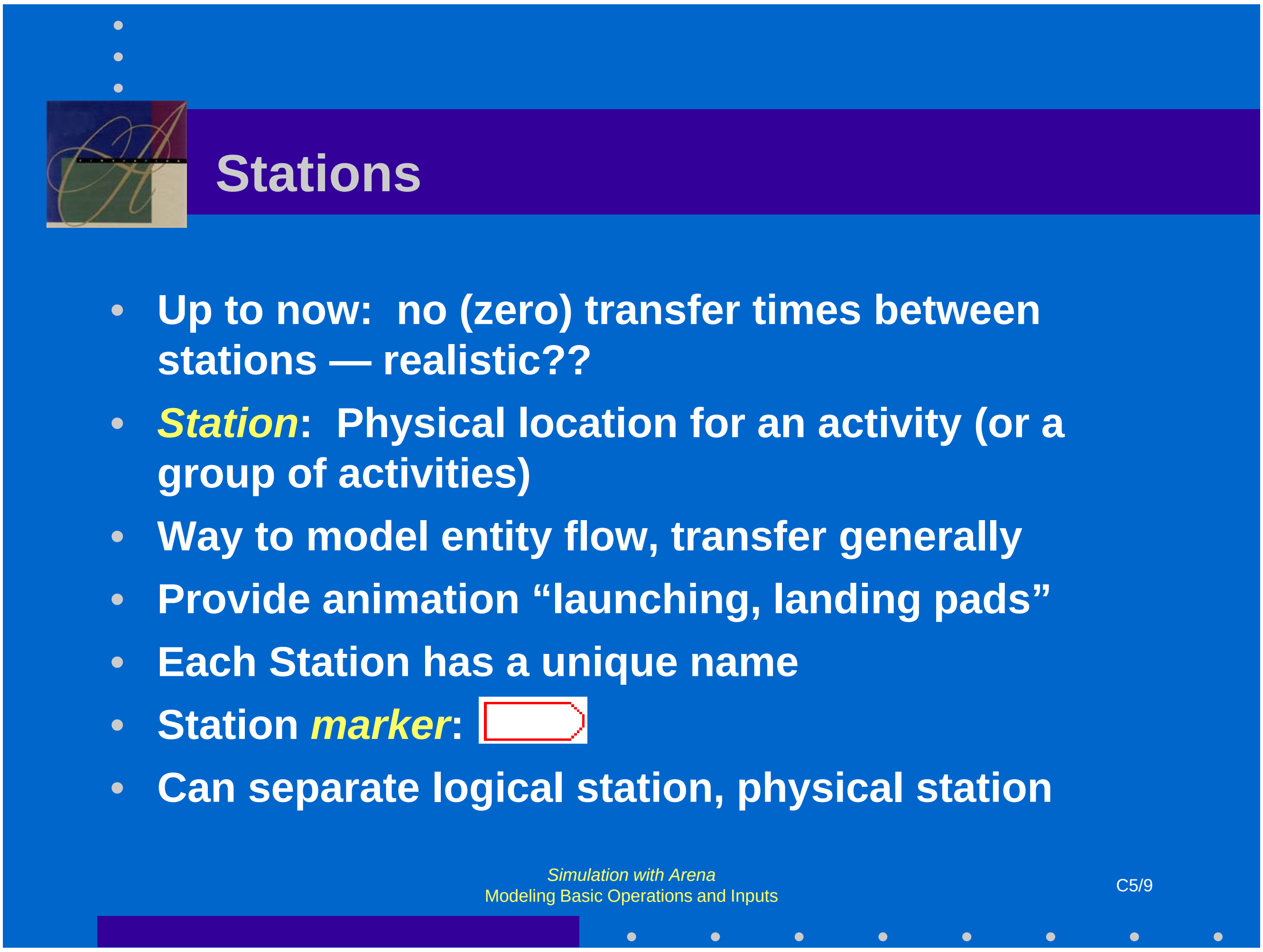
Run Conditions, Output, Animation

- **Start empty & idle, run for 2,000 minutes**
- **Output:**
  - Utilization of all resources
  - Number in each queue
  - Time in each queue
  - Cycle time (flowtime) separated out by shipped parts, salvaged/shipped parts, scrapped parts
- **Animation:**
  - Queues, busy/idle resources as before
  - Entity movement between stations (2 min. transfers)



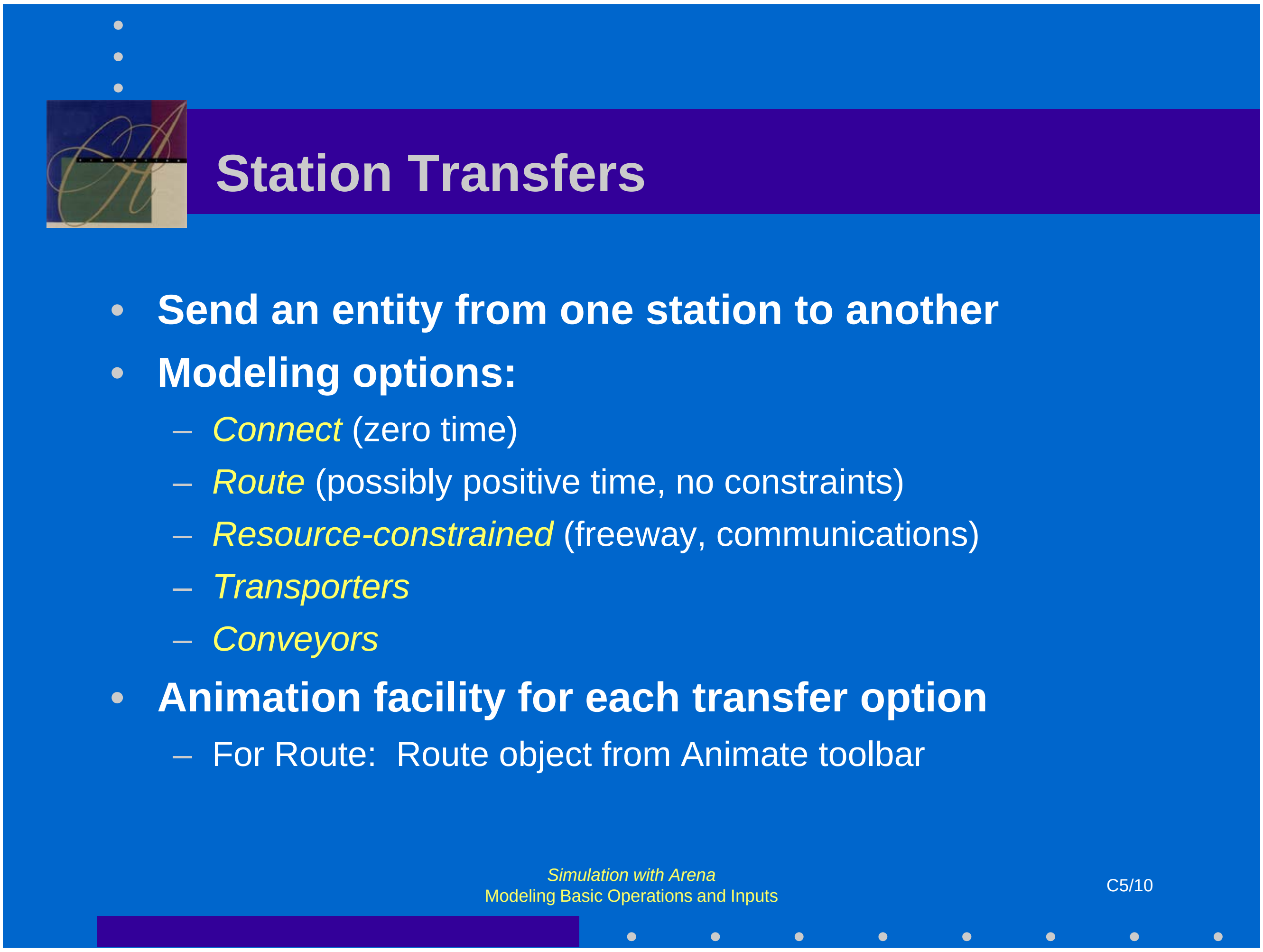
Developing a Modeling Approach

- **Define submodels, modules, data structures, control logic**
- **Appropriate level of detail — judgment call**
- **Often multiple ways to model, represent logic**
- **This model:**
  - Separate Arrive modules for two part types
  - Separate Server modules for each Prep area
  - Inspect modules for Sealer and Rework
  - Depart modules for Shipping, Salvage, Scrap
  - Transfer times: *Route*
  - Attribute **`Sealer Time`** assigned at Arrival (parts have *different* times at *the* Sealer station)



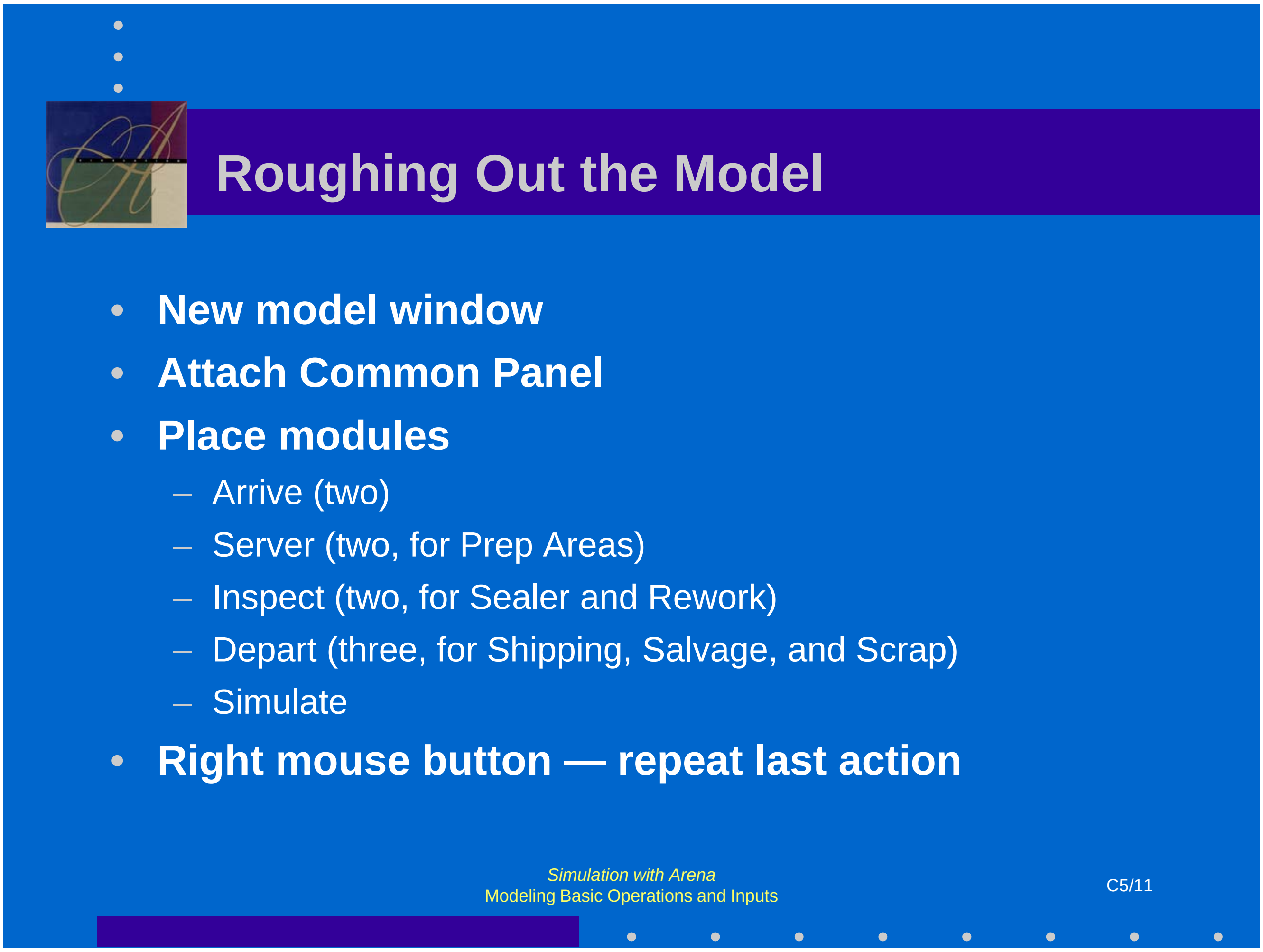
Stations

- Up to now: no (zero) transfer times between stations — realistic??
- *Station*: Physical location for an activity (or a group of activities)
- Way to model entity flow, transfer generally
- Provide animation “launching, landing pads”
- Each Station has a unique name
- Station *marker*: ▭
- Can separate logical station, physical station



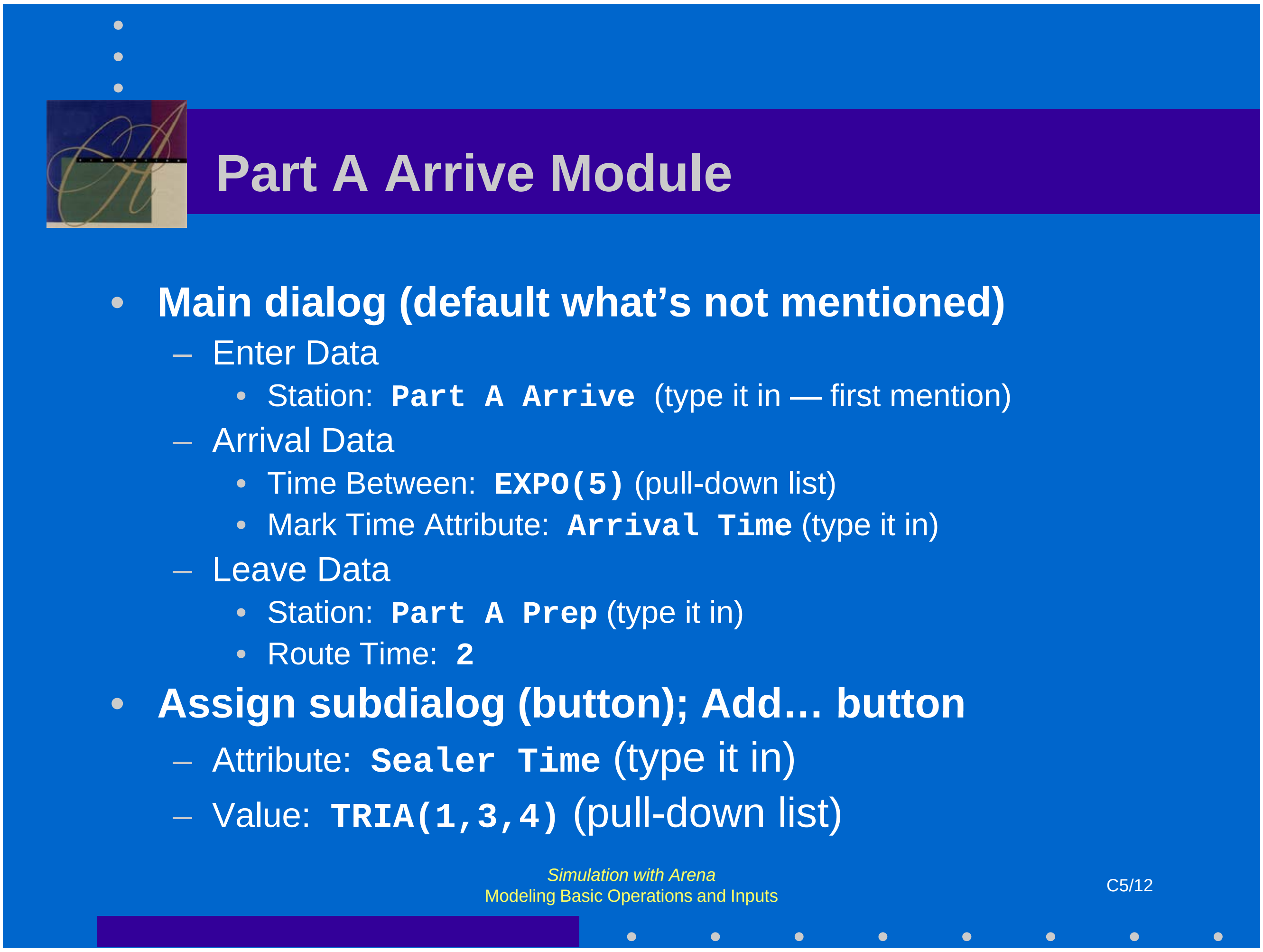
Station Transfers

- Send an entity from one station to another
- Modeling options:
  - *Connect* (zero time)
  - *Route* (possibly positive time, no constraints)
  - *Resource-constrained* (freeway, communications)
  - *Transporters*
  - *Conveyors*
- Animation facility for each transfer option
  - For Route: Route object from Animate toolbar



Roughing Out the Model

- **New model window**
- **Attach Common Panel**
- **Place modules**
  - Arrive (two)
  - Server (two, for Prep Areas)
  - Inspect (two, for Sealer and Rework)
  - Depart (three, for Shipping, Salvage, and Scrap)
  - Simulate
- **Right mouse button — repeat last action**



Part A Arrive Module

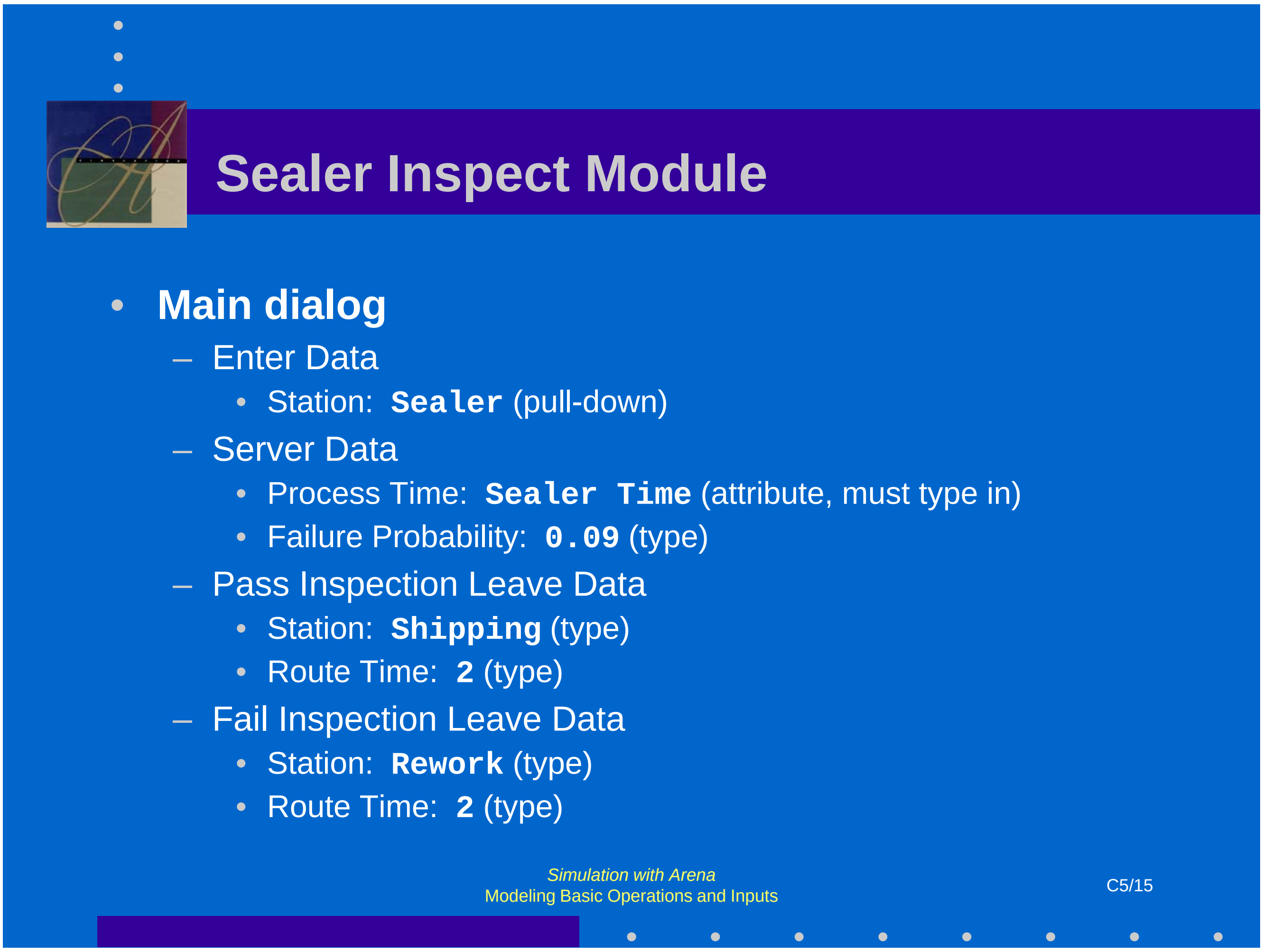
- **Main dialog (default what’s not mentioned)**
  - Enter Data
    - Station: **`Part A Arrive`** (type it in — first mention)
  - Arrival Data
    - Time Between: **`EXPO(5)`** (pull-down list)
    - Mark Time Attribute: **`Arrival Time`** (type it in)
  - Leave Data
    - Station: **`Part A Prep`** (type it in)
    - Route Time: **`2`**
- **Assign subdialog (button); Add… button**
  - Attribute: **`Sealer Time`** (type it in)
  - Value: **`TRIA(1,3,4)`** (pull-down list)

Part B Arrive Module

- **Same as for Part A Arrive, except:**
  - Station: **`Part B Arrive`**
  - Batch Size: **`4`**
  - Time Between: **`EXPO(30)`**
  - Leave Data Station: **`Part B Prep`**
  - **`Sealer Time`** Attrib. Value: **`NORM(2.4, 0.5)`**
- **Each arrival creates four separate entities**
  - Quadruplets separated at birth
  - Flow independently
  - Independent **`Sealer Time`** values assigned

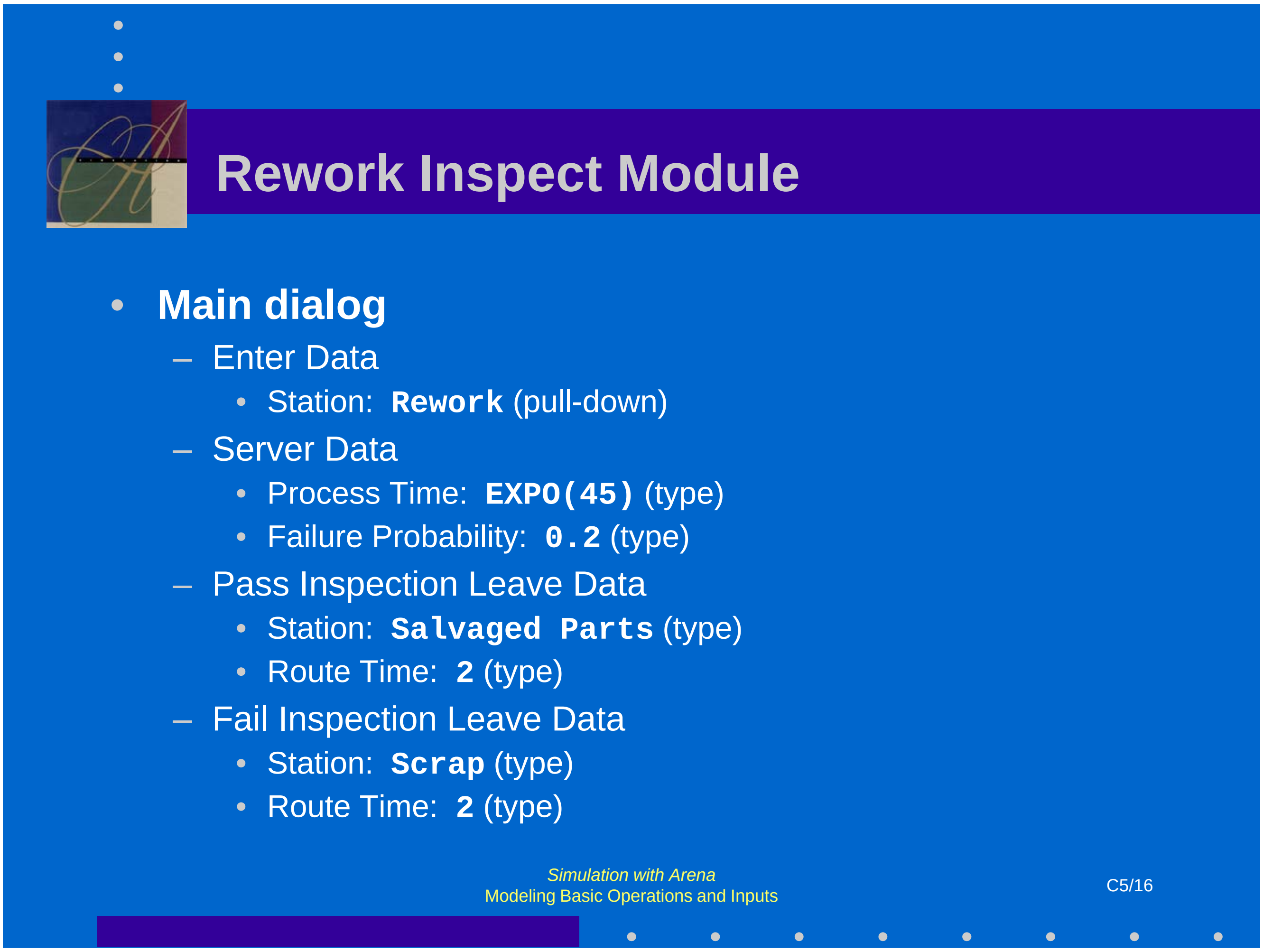
Part A [B] Prep Server Modules

- **Exploit pull-downs where possible (Station names, Attribute names) for earlier definitions**
- **Main dialog**
  - Enter Data
    - Station: **`Part A Prep`** [**`Part B Prep`**] (pull-down)
  - Server Data
    - Process Time: **`TRIA(1,4,8)`** [**`TRIA(3,5,10)`**]
  - Leave Data
    - Station: **`Sealer`** (type)
    - Route Time: **`2`**
- **Accept defaults for Resource name, Resource Statistics, subdialogs (Queue, Resource, etc.)**



Sealer Inspect Module

- **Main dialog**
  - Enter Data
    - Station: **`Sealer`** (pull-down)
  - Server Data
    - Process Time: **`Sealer Time`** (attribute, must type in)
    - Failure Probability: **`0.09`** (type)
  - Pass Inspection Leave Data
    - Station: **`Shipping`** (type)
    - Route Time: **`2`** (type)
  - Fail Inspection Leave Data
    - Station: **`Rework`** (type)
    - Route Time: **`2`** (type)



Rework Inspect Module

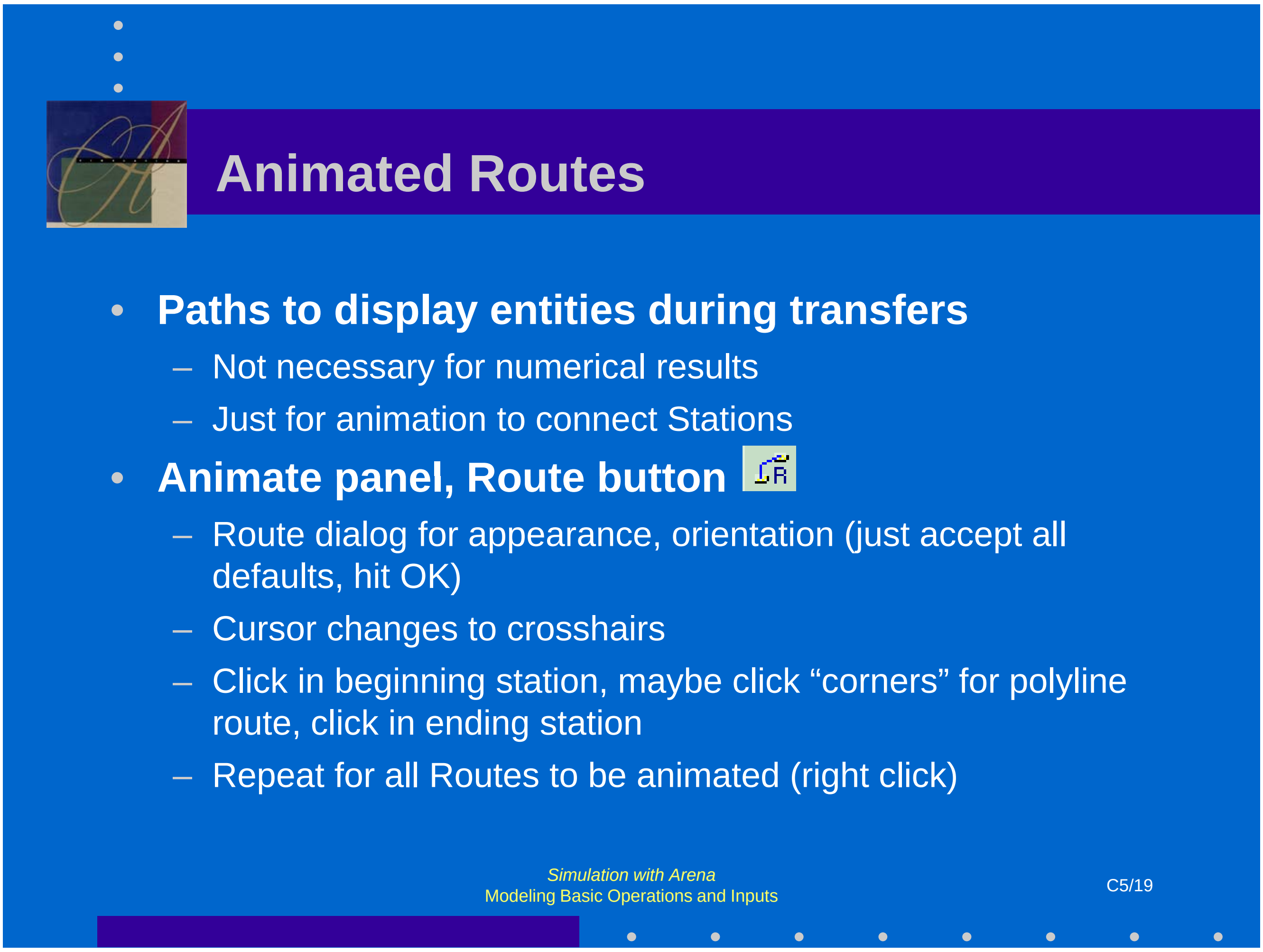
- **Main dialog**
  - Enter Data
    - Station: **`Rework`** (pull-down)
  - Server Data
    - Process Time: **`EXPO(45)`** (type)
    - Failure Probability: **`0.2`** (type)
  - Pass Inspection Leave Data
    - Station: **`Salvaged Parts`** (type)
    - Route Time: **`2`** (type)
  - Fail Inspection Leave Data
    - Station: **`Scrap`** (type)
    - Route Time: **`2`** (type)

Depart Modules

- **Three separate modules: `Shipping`, `Salvaged Parts`, and `Scrap`**
- **Main dialog for `Shipping` (others are similar)**
  - Enter Data
    - Station: **`Shipping`** (pull-down)
  - Count
    - Individual Counter: *select* (accept default counter name)
  - Tally
    - Individual Tally: *select*
    - Attribute: **`Arrival Time`** (accept Interval default selection)

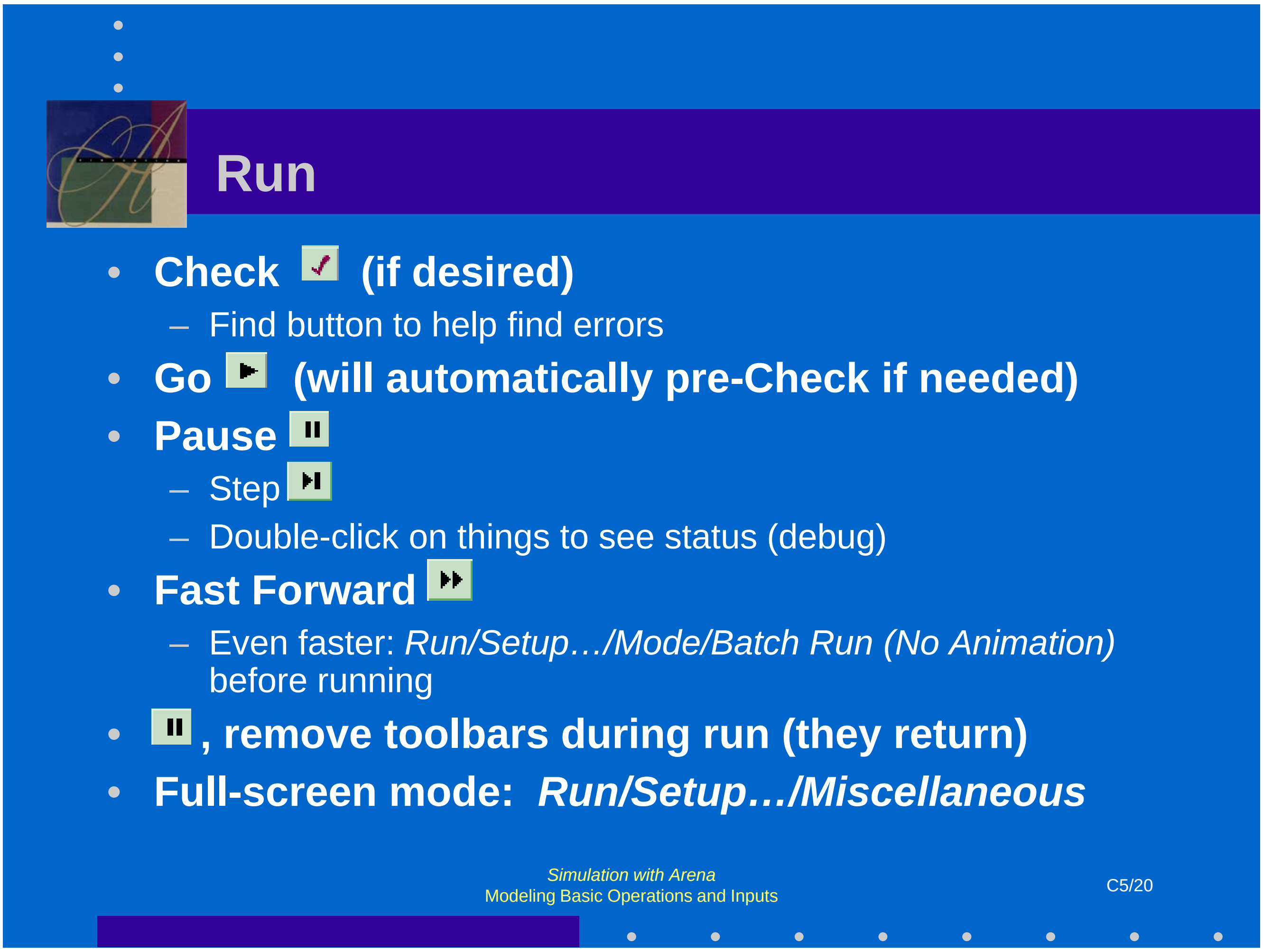
Simulate Module

- **Specify termination rule (among other things)**
- **Main dialog**
  - Project
    - Title: **`Electronic Assembly and Test`**
    - Analyst: **`Mr. Munchkin`**
  - Replicate
    - Length of Replication: **`2000`**
- **Accept defaults for**
  - Date (computer clock)
  - Number of Replications (1)
  - Beginning Time (0.0)
  - Initialize everything between replications



Animated Routes

- **Paths to display entities during transfers**
  - Not necessary for numerical results
  - Just for animation to connect Stations
- **Animate panel, Route button** ⤴R
  - Route dialog for appearance, orientation (just accept all defaults, hit OK)
  - Cursor changes to crosshairs
  - Click in beginning station, maybe click “corners” for polyline route, click in ending station
  - Repeat for all Routes to be animated (right click)



Run

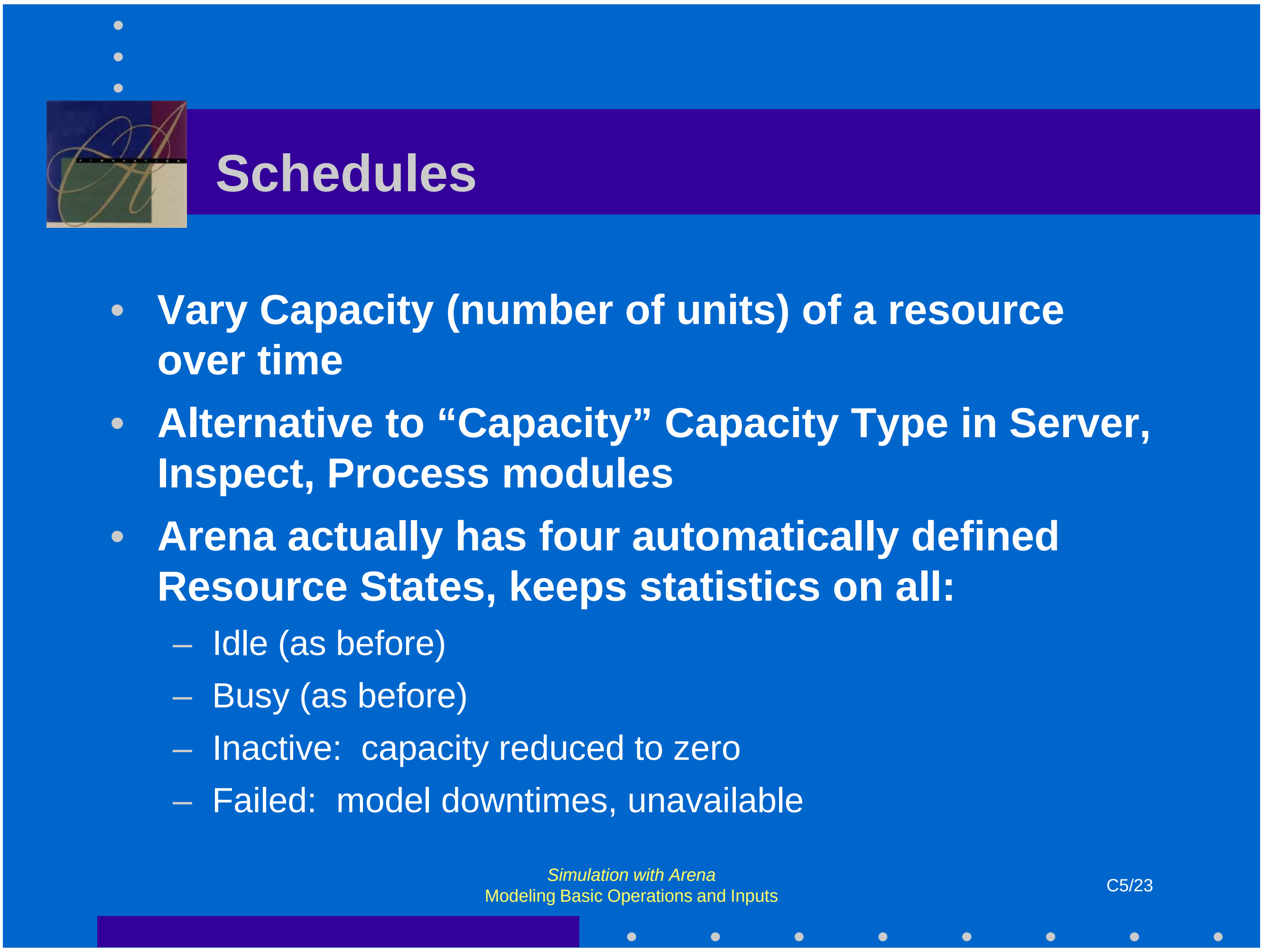
- **Check** ✓ **(if desired)**
  - Find button to help find errors
- **Go** ▶ **(will automatically pre-Check if needed)**
- **Pause** ⏸
  - Step ⏭
  - Double-click on things to see status (debug)
- **Fast Forward** ⏩
  - Even faster: *Run/Setup…/Mode/Batch Run (No Animation)* before running
- ⏸**, remove toolbars during run (they return)**
- **Full-screen mode: *Run/Setup…/Miscellaneous***

Viewing the Results

- **When done, asked if you want to see numerical results (text)**
  - Uses Notepad or other viewer in separate window
  - Also saves as text file **`model_name.out`**
- **Tally, Discrete Change (a.k.a. time-persistent), Counters areas (if present in model)**
  - Columns for averages, min, max, number of observations or final value
- **Half Width column:**
  - For 95% confidence interval on *steady-state* (*long-run*) expected average
  - May not have enough data (see Chapter 7 ...)

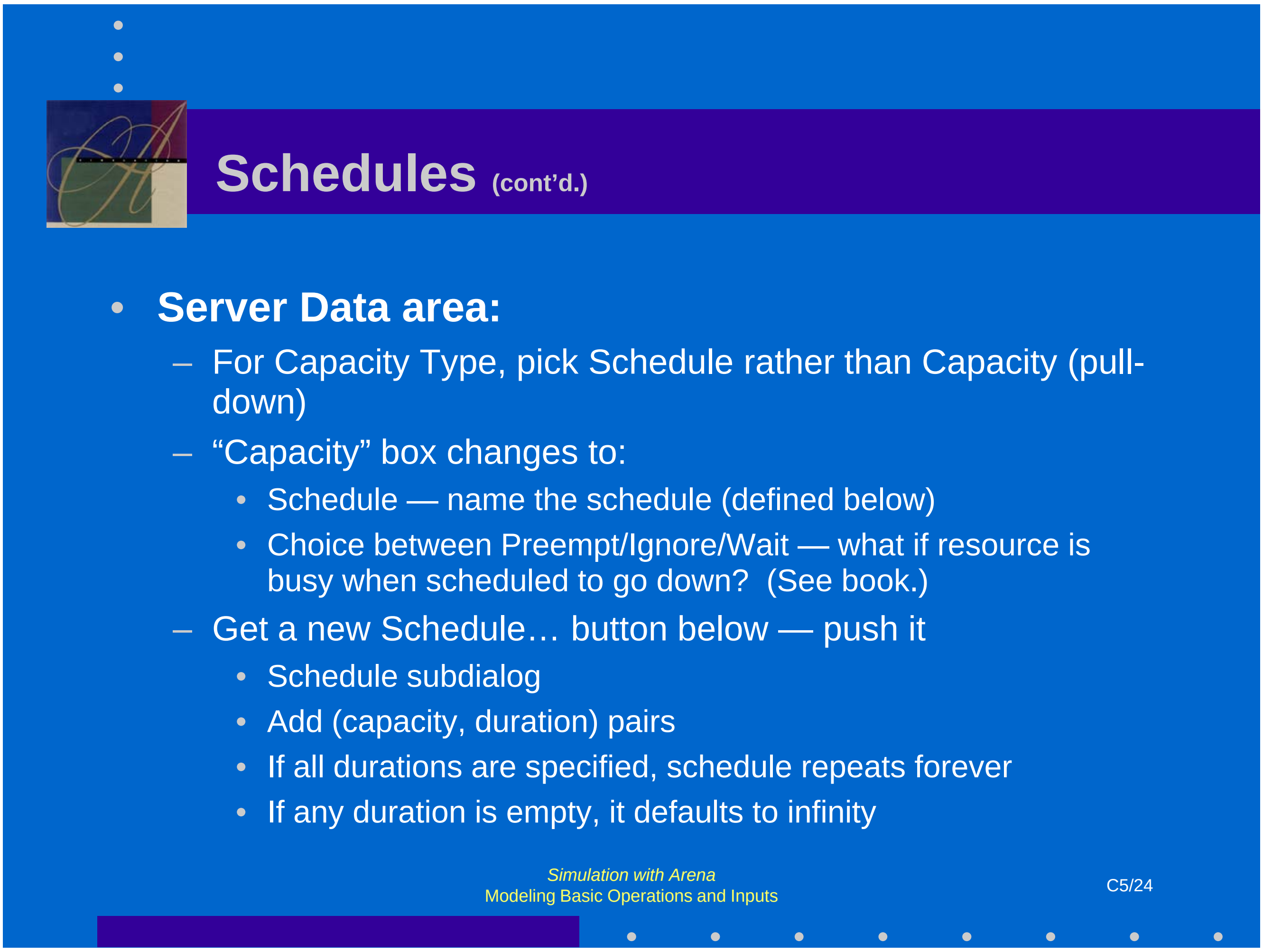
Enhanced Model (Model 5.2)

- **A Story**
  - Original model shown to production manager
  - Pointed out that this is only the first shift of a two-shift day — on second shift there are two operators at Rework (the bottleneck station)
  - Pointed out that the Sealer fails sometimes
    - Uptimes ~ exponential, mean 2 hours
    - Repair times ~ exponential, mean 4 minutes
- **Need: *Schedules*, *Resource States*, *Resource Failures***



Schedules

- **Vary Capacity (number of units) of a resource over time**
- **Alternative to “Capacity” Capacity Type in Server, Inspect, Process modules**
- **Arena actually has four automatically defined Resource States, keeps statistics on all:**
  - Idle (as before)
  - Busy (as before)
  - Inactive: capacity reduced to zero
  - Failed: model downtimes, unavailable



Schedules (cont’d.)

- **Server Data area:**
  - For Capacity Type, pick Schedule rather than Capacity (pull-down)
  - “Capacity” box changes to:
    - Schedule — name the schedule (defined below)
    - Choice between Preempt/Ignore/Wait — what if resource is busy when scheduled to go down? (See book.)
  - Get a new Schedule… button below — push it
    - Schedule subdialog
    - Add (capacity, duration) pairs
    - If all durations are specified, schedule repeats forever
    - If any duration is empty, it defaults to infinity

Resource Downtimes

- **Bring one unit of a resource down — other units (if any) still up**
- **Resource… button**
  - Downtime Name
  - Time Between Downtimes (anything — pull-down for distributions)
  - Downtime (anything, distribution pull-down)
- **Can have multiple Downtimes (separate names) for a Resource**

Resource Failures

- **All units of a resource come down**
- **Resource… button**
  - Failure Name
  - Based on entity Count or elapsed Time
  - Preempt/Ignore/Wait for come-down rule
  - If based on Count, the Count for uptime

    If based on Time, the Uptime
  - Downtime (anything, distribution pull-down)
- **Can have multiple Failures (separate names) for a Resource**

Saving Statistical Data

- **Observe, maybe save different kinds of data**
  - Non-default output performance measure
    - e.g., % of time queue length > 5
  - Postprocessing via Output Analyzer
    - Note that dynamic animated plots disappear when done
    - Statistical analysis of output data, statistical inference
  - Export to other applications (spreadsheets, etc.)
- **Save records of Time-Persistent data, Tallies, Counters, “Frequencies” (new)**
- **How? *Statistics module* (Common panel)**

The Statistics Module

- **Five different areas, for different kinds of stats**
- **In an area, Add… button for what you want**
  - Subdialog depends on type area (type of stat)
  - Option to save data to a (binary) file — **`“name.dat”`** (including the double quotes); name could include drive, path
- **Time-Persistent area**
  - Select data object, later dialogs react to selection
- **Tallies area**
  - Select Tally Name
- **Other areas discussed later ...**

Frequency Statistics

- **A "finer" description of an output**
- **Record time-persistent occurrence frequency of a Variable, Expression, or State**
- **Example: Want to know % of time the Rework queue is of length 0, (0, 10], (10, 20], etc.**
  - Statistics module, Frequencies area
  - Add… button
  - Expression: Variable, general expression
    - Arena function NQ(queue name): queue length
    - Others: NR(resource name): no. busy
      MR(resource name): no. available
  - Define categories (Constant or Range)

The Output Analyzer

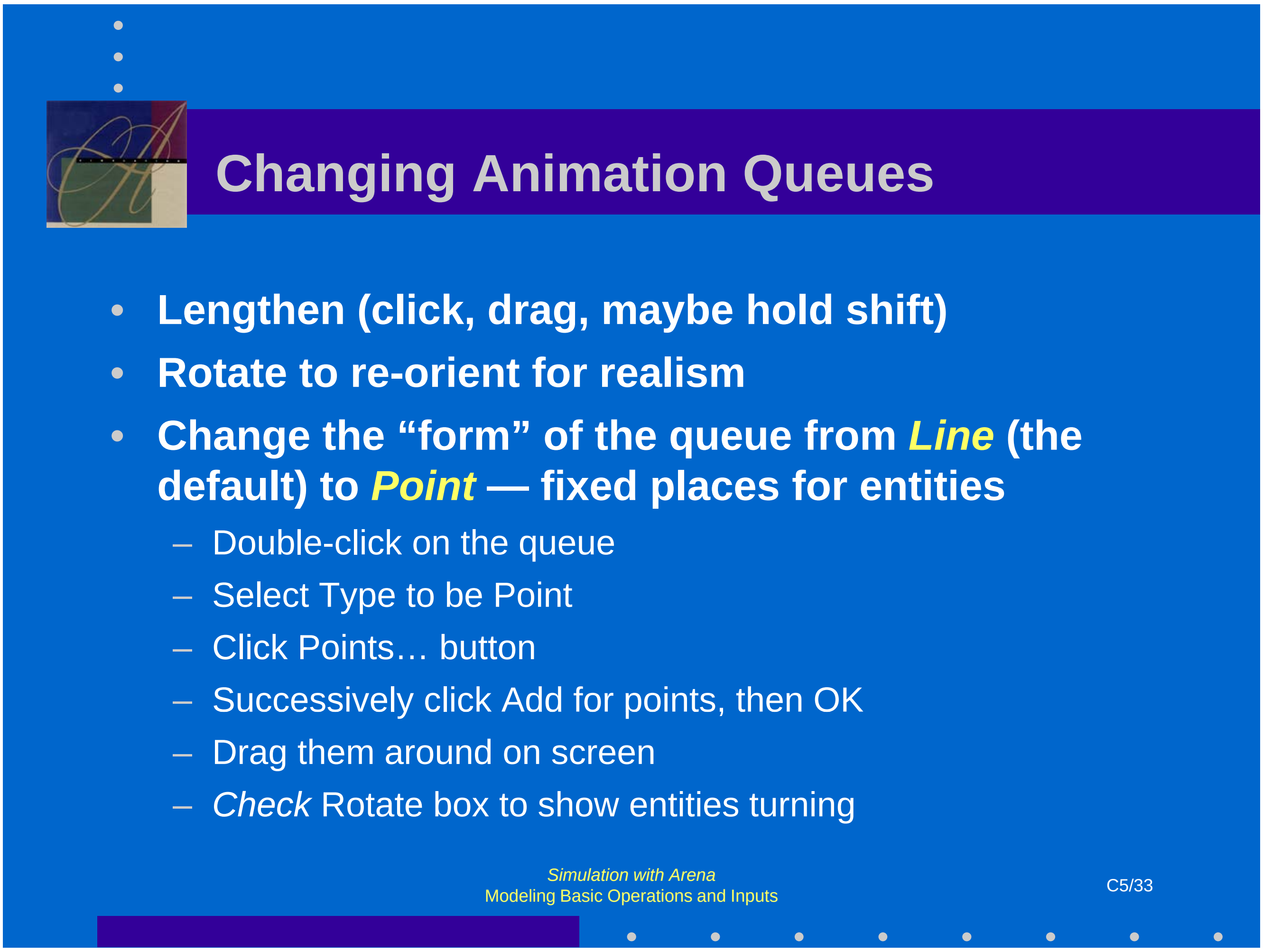
- **Separate application, also accessible via Tools menu in Arena**
- **Reads binary files saved by Arena**
- **Various kinds of output-data display, analysis**
  - For now: just data-display functions
- **Advisable (not required) — define, maybe save a *data group* (*File/New* or 🗋 , then Add…)**
  - List of output files of interest — one model or many
  - Eases tasks by "screening" for these files only
  - Save in file called `whatever.dgr`, Open next time

The Output Analyzer (cont'd.)

- ***Plot*** **time-persistent data**
  - *Graph/Plot* or
  - Can overlay several curves (Sensible? Units?)
  - Options for plot Title, axis Labels, crop axes
- ***Moving-average plots*** **: "smooth" over time**
  - Moving-average window Value
  - Exponential smoothing, Forecasting
- ***Barcharts*** **: like Plot, cosmetically different**
- ***Histograms*** **of data**
  - *Beware*: autocorrelation

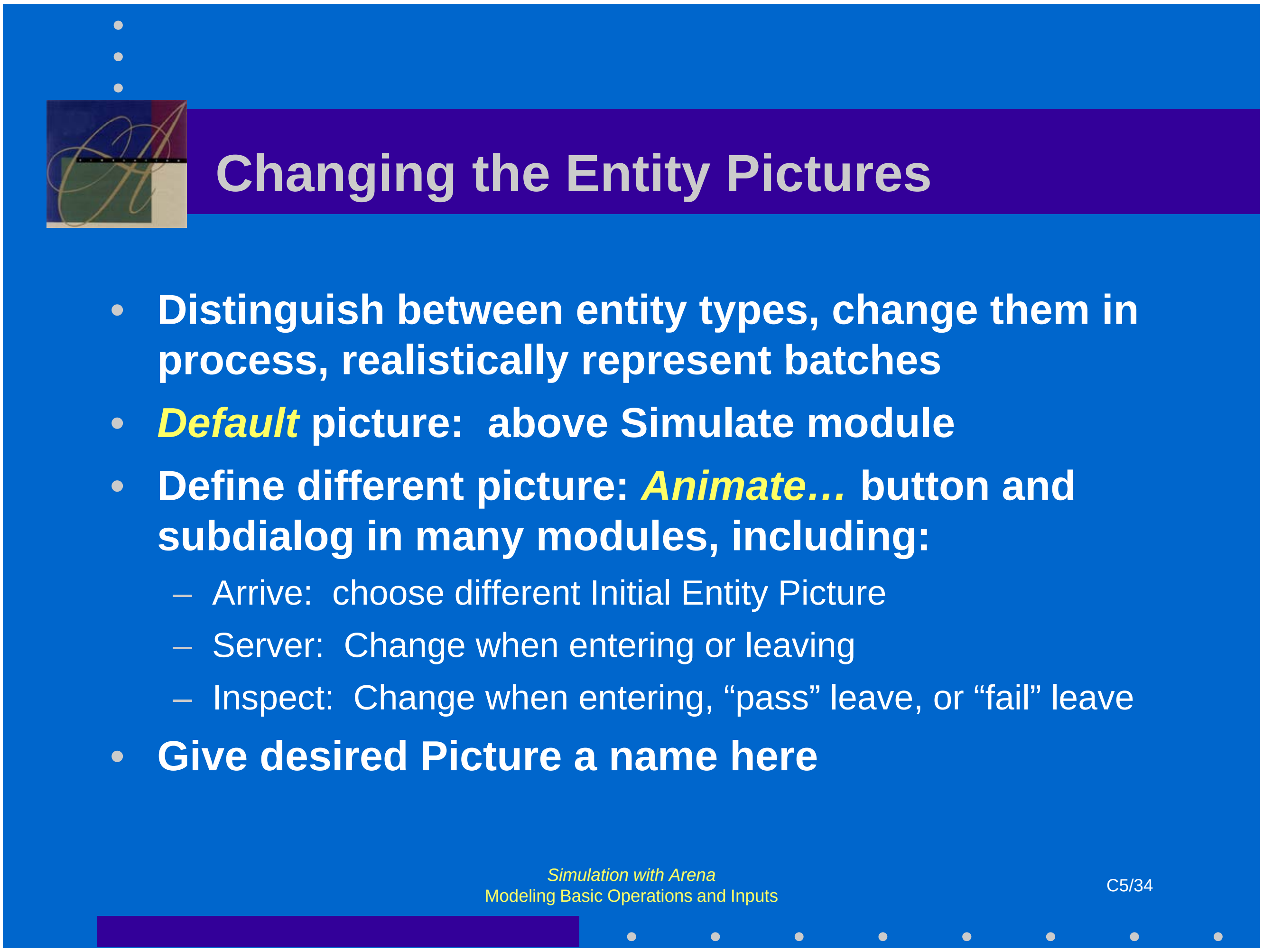
Enhancing the Animation (Model 5.3)

- **Get "Spartan" generic default animation for many things**
  - Usually sufficient for verification, validation
- **Often want to customize, enhance it a bit**
  - More realism, impact
- **Option to pull animation away from model logic in model window**
  - Useful for big models, complex animation
  - Set up Named Views for model logic, animation, or close-ups of parts of animation



Changing Animation Queues

- **Lengthen (click, drag, maybe hold shift)**
- **Rotate to re-orient for realism**
- **Change the “form” of the queue from *Line* (the default) to *Point* — fixed places for entities**
  - Double-click on the queue
  - Select Type to be Point
  - Click Points… button
  - Successively click Add for points, then OK
  - Drag them around on screen
  - *Check* Rotate box to show entities turning



Changing the Entity Pictures

- **Distinguish between entity types, change them in process, realistically represent batches**
- ***Default* picture: above Simulate module**
- **Define different picture: *Animate…* button and subdialog in many modules, including:**
  - Arrive: choose different Initial Entity Picture
  - Server: Change when entering or leaving
  - Inspect: Change when entering, “pass” leave, or “fail” leave
- **Give desired Picture a name here**

Changing the Entity Pictures (cont'd.)

- **After defining names, must edit/create/read**
  - Double-click on default picture above Simulate
  - Make sure Default picture is selected (depressed)
  - Copy, select the copied picture
  - Select name from Value pull-down to rename copy
  - Either
    - Double-click to edit (for artists only)
    - Open Picture library (**`.plb`** file), select desired picture from scrolling window, hit << button
- **Reference point — where entity moves, sits**
- **Application: "hidden" batches (Model 5.3)**

Changing Resource Pictures

- **Realism, indicate state (Idle, Busy, etc.)**
- **Double-click, edit similarly to entity pictures**
  - Artwork
  - Picture libraries (**`.plb`** files)
  - Example: Sealer resource in Model 5.3
- **Seize point — place for realism (layers, etc.)**
- **Adjust size — Size Factor**
- **Multiple-capacity resources**
  - Multiple seize points (Rework resource, Model 5.3)

Adding Plots and Animated Variables

- **Animate module from Common panel**
  - Alternative: Animate toolbar buttons, but Animate module is easier
- **Select Data Object to observe**
- **Select Information to display (depends on Data Object selected)**
- **Check off mode(s) of display (default: all)**
  - For Plots:
    - Have to guess at Max Y (maybe revise after run …)
    - History Points = no. of plot points to display at a time

Input Analysis: Specifying Model Parameters, Distributions

- ***Structural*** **modeling: what we’ve done so far**
  - Logical aspects — entities, resources, paths, etc.
- ***Quantitative*** **modeling**
  - Numerical, distributional specifications
  - Like structural modeling, need to observe system’s operation, take data if possible

Deterministic vs. Random Inputs

- ***Deterministic*: nonrandom, fixed values**
  - Number of units of a resource
  - Entity transfer time (?)
  - Interarrival, processing times (?)
- ***Random* (a.k.a. *stochastic*): model as a distribution, “draw” or “generate” values from to drive simulation**
  - Transfer, Interarrival, Processing times
  - What distribution? What distributional parameters?
  - Causes simulation output to be random, too
- **Don’t just assume randomness away — validity**

Collecting Data

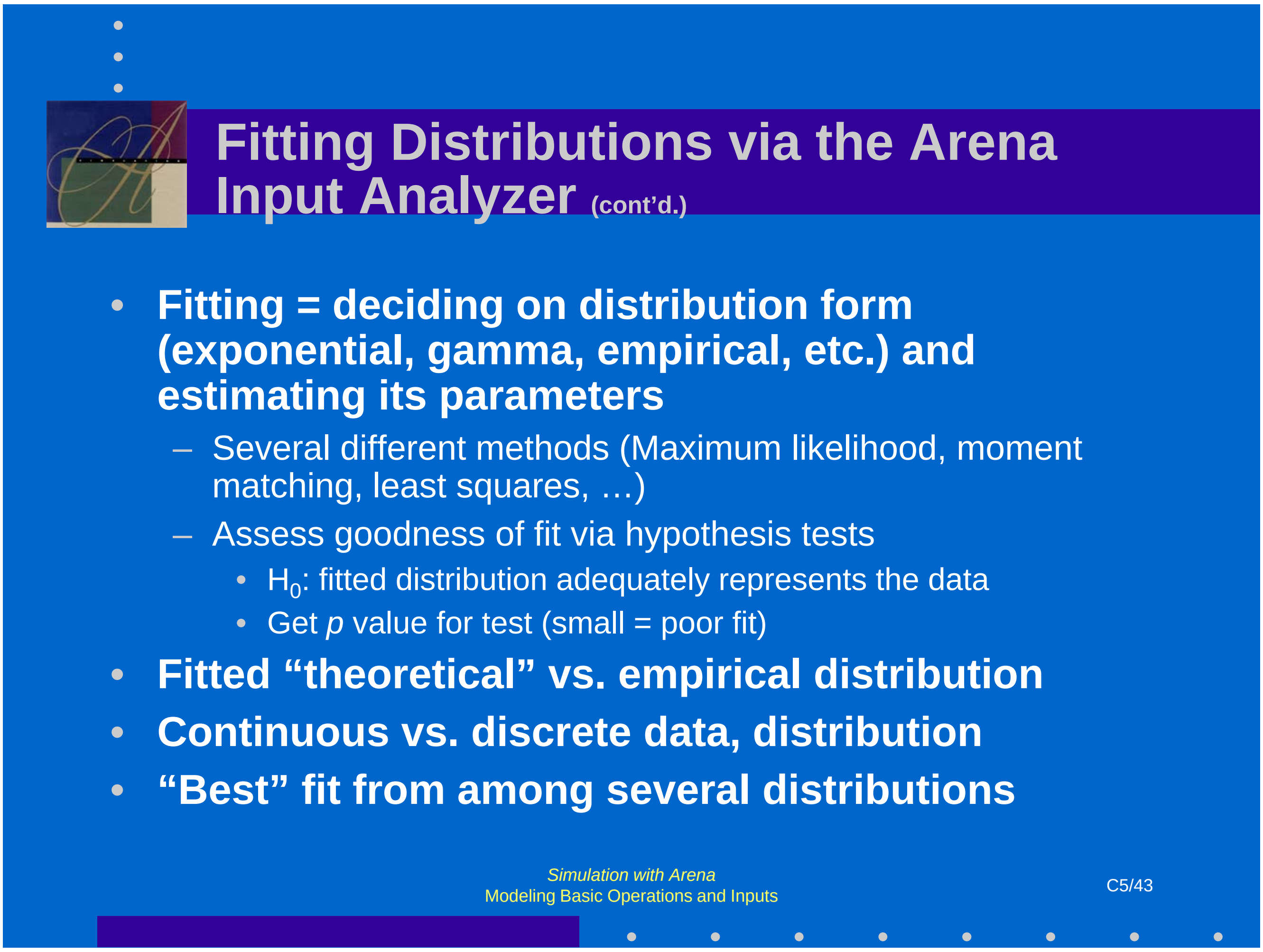
- **Generally hard, expensive, frustrating, boring**
  - System might not exist
  - Data available on the wrong things — might have to change model according to what’s available
  - Incomplete, “dirty” data
  - Too much data (!)
- **Sensitivity of outputs to uncertainty in inputs**
- **Match model detail to quality of data**
- **Cost — should be budgeted in project**
- **Capture variability in data — model validity**
- **Garbage In, Garbage Out (GIGO)**

Using Data: Alternatives and Issues

- **Use data "directly" in simulation**
  - Read actual observed values to drive the model inputs (interarrivals, service times, part types, …)
  - All values will be "legal" and realistic
  - But can never go outside your observed data
  - May not have enough data for long or many runs
  - Computationally slow (reading disk files)
- **Or, fit probability distribution to data**
  - "Draw" or "generate" synthetic observations from this distribution to drive the model inputs
  - We've done it this way so far
  - Can go beyond observed data (good and bad)
  - May not get a good "fit" to data — validity?

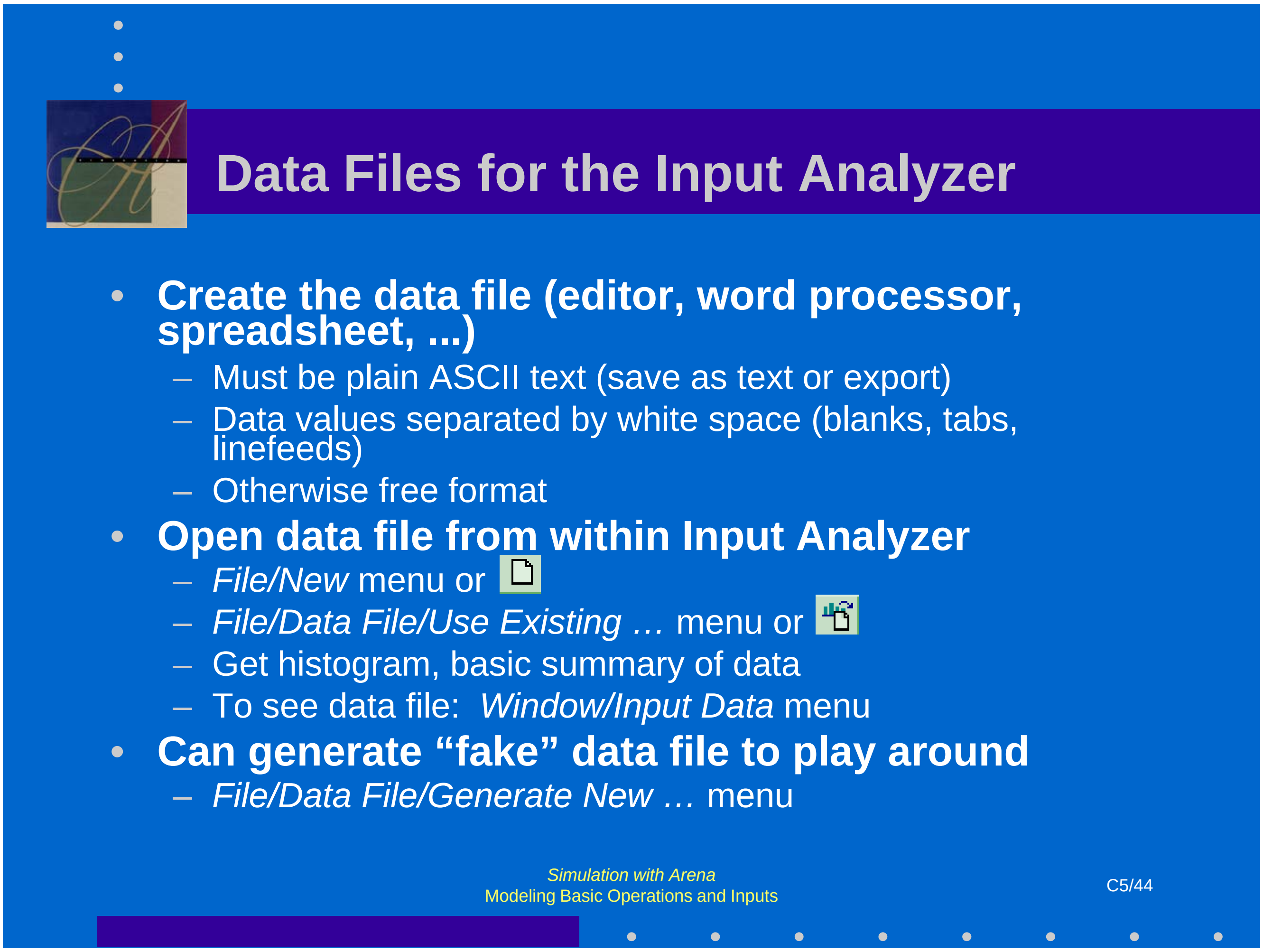
Fitting Distributions via the Arena Input Analyzer

- **Assume:**
  - Have sample data: Independent and Identically Distributed (IID) list of observed values from the actual physical system
  - Want to select or fit a probability distribution for use in generating inputs for the simulation model
- **Arena Input Analyzer**
  - Separate application, also accessible via Tools menu in Arena
  - Fits distributions, gives valid Arena expression for generation to paste directly into simulation model



Fitting Distributions via the Arena Input Analyzer (cont'd.)

- **Fitting = deciding on distribution form (exponential, gamma, empirical, etc.) and estimating its parameters**
  - Several different methods (Maximum likelihood, moment matching, least squares, …)
  - Assess goodness of fit via hypothesis tests
    - H_0: fitted distribution adequately represents the data
    - Get *p* value for test (small = poor fit)
- **Fitted "theoretical" vs. empirical distribution**
- **Continuous vs. discrete data, distribution**
- **"Best" fit from among several distributions**



Data Files for the Input Analyzer

- **Create the data file (editor, word processor, spreadsheet, ...)**
  - Must be plain ASCII text (save as text or export)
  - Data values separated by white space (blanks, tabs, linefeeds)
  - Otherwise free format
- **Open data file from within Input Analyzer**
  - *File/New* menu or 🗋
  - *File/Data File/Use Existing …* menu or 🗋
  - Get histogram, basic summary of data
  - To see data file: *Window/Input Data* menu
- **Can generate "fake" data file to play around**
  - *File/Data File/Generate New …* menu

The Fit Menu

- **Fits distributions, does goodness-of-fit tests**
- **Fit a specific distribution form**
  - Plots density over histogram for visual “test”
  - Gives exact expression to Copy and Paste (Ctrl+C, Ctrl+V) over into simulation model
  - May include “offset” depending on distribution
  - Gives results of goodness-of-fit tests
    - Chi square, Kolmogorov-Smirnov tests
    - Most important part: *p-value*, always between 0 and 1:

      Probability of getting a data set that’s more inconsistent with the fitted distribution than the data set you actually have, if the the fitted distribution is truly “the truth”

      “Small” *p* (< 0.05 or so): poor fit (try again or give up)

The Fit Menu (cont’d.)

- **Fit all Arena’s (theoretical) distributions at once**
  - *Fit/Fit All* menu or
  - Returns the *minimum square-error* distribution
    - Square error = sum of squared discrepancies between histogram frequencies and fitted-distribution frequencies
    - Can depend on histogram intervals chosen: different intervals can lead to different “best” distribution
  - Could still be a poor fit, though (check *p* value)
  - To see all distributions, ranked: *Window/Fit All Summary* or

The Fit Menu (cont'd.)

- **"Fit" Empirical distribution (continuous or discrete): *Fit/Empirical***
  - Can interpret results as a Discrete or Continuous distribution
    - Discrete: get pairs (*Cumulative* Probability, Value)
    - Continuous: Arena will linearly interpolate *within* the data range according to these pairs (so you can never generate values outside the range, which might be good or bad)
  - Empirical distribution can be used when "theoretical" distributions fit poorly, or intentionally

Some Issues in Fitting Input Distributions

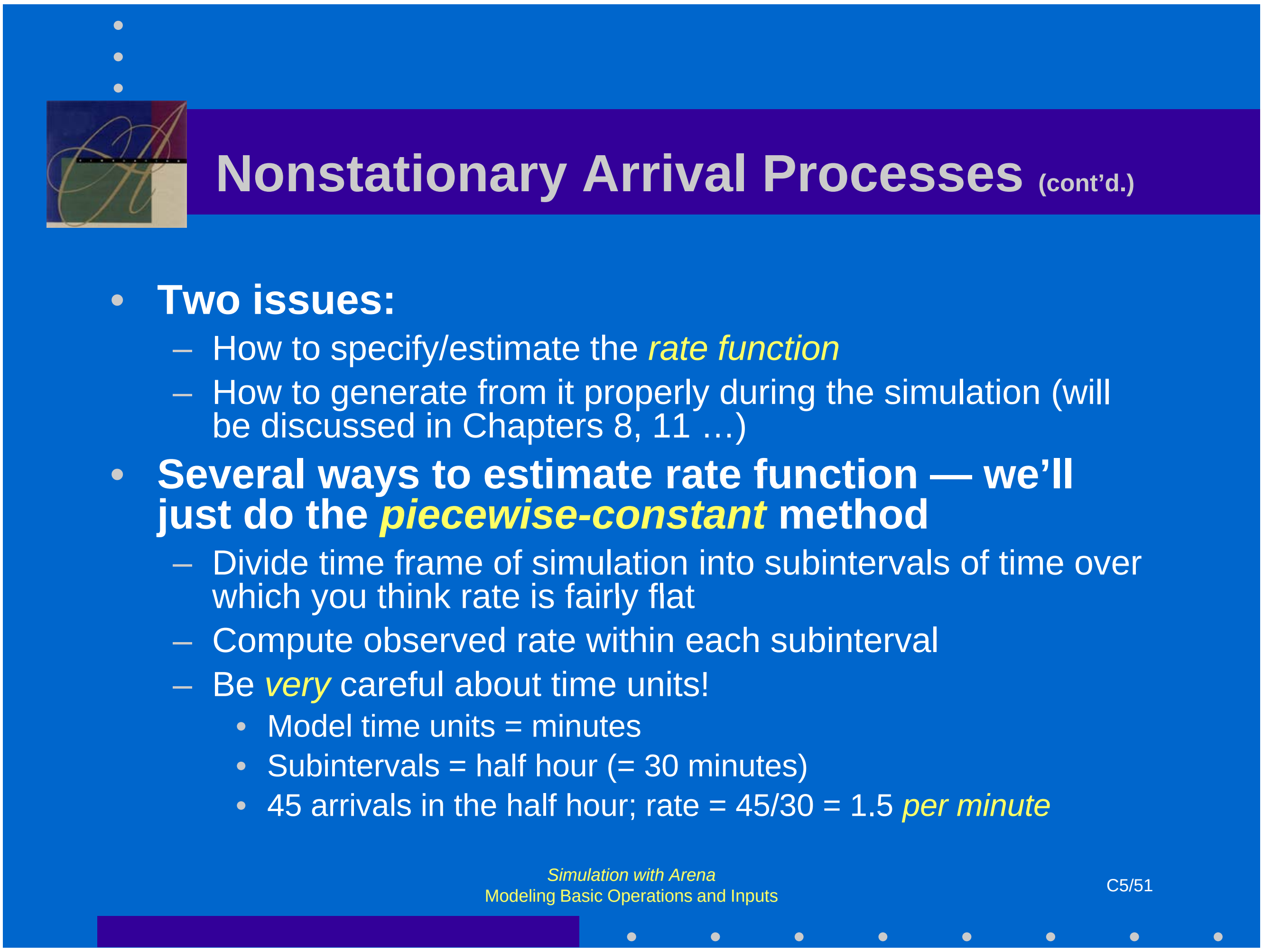
- **Not an exact science — no "right" answer**
- **Consider theoretical vs. empirical**
- **Consider range of distribution**
  - Infinite both ways (e.g., normal)
  - Positive (e.g., exponential, gamma)
  - Bounded (e.g., beta, uniform)
- **Consider ease of parameter manipulation to affect means, variances**
- **Simulation model sensitivity analysis**
- **Outliers, multimodal data**
  - Maybe split data set (see textbook for details)

No Data?

- **Happens more often than you'd like**
- **No good solution; some (bad) options:**
  - Interview "experts"
    - Min, Max: Uniform
    - Avg., % error or absolute error: Uniform
    - Min, Mode, Max: Triangular
      - Mode can be different from Mean — allows asymmetry
  - Interarrivals — independent, stationary
    - Exponential — still need some value for mean
  - Number of "random" events in an interval: Poisson
  - Sum of independent "pieces": normal
  - Product of independent "pieces": lognormal

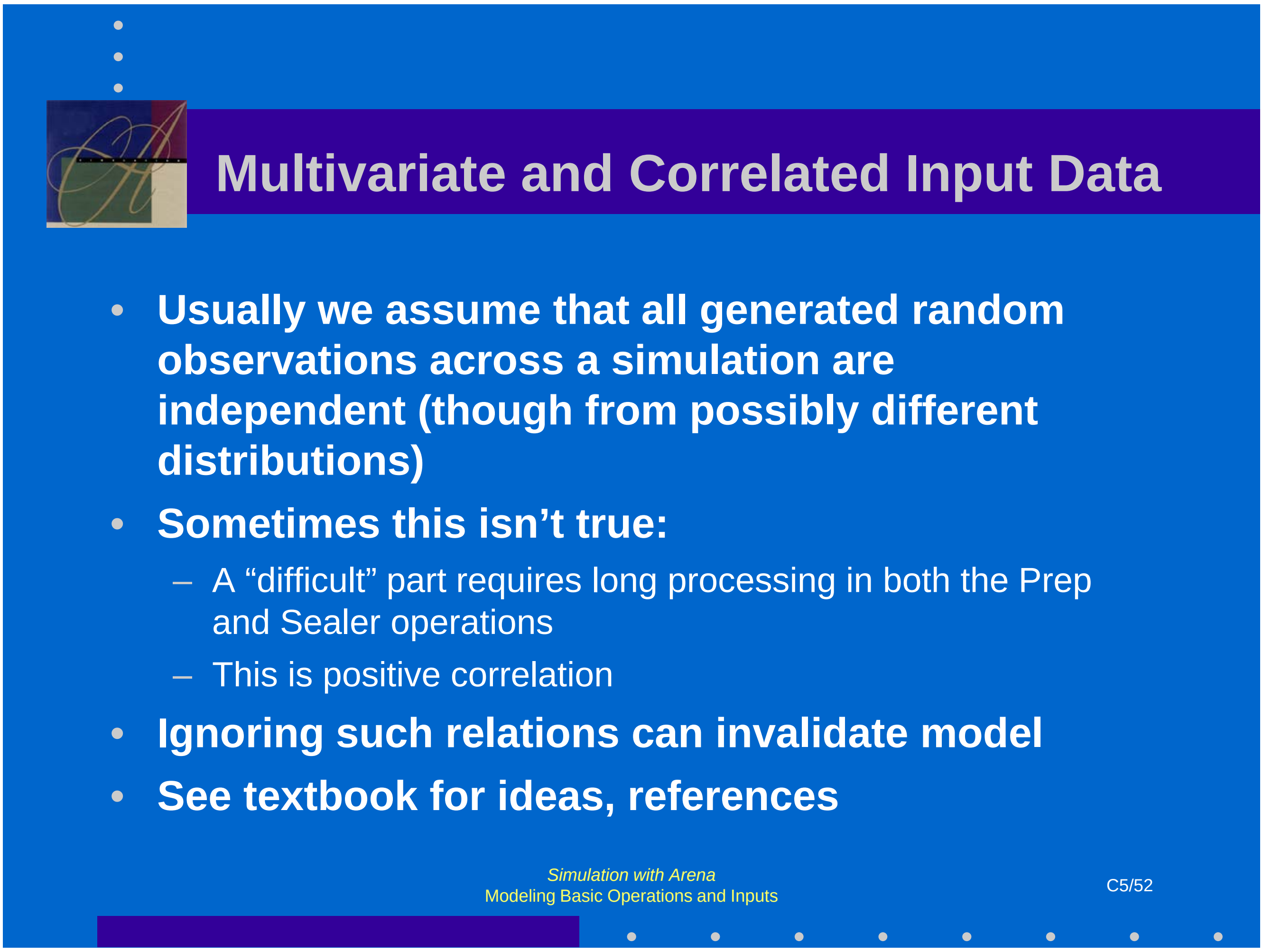
Nonstationary Arrival Processes

- **External events (often arrivals) whose rate varies over time**
  - Lunchtime at fast-food restaurants
  - Rush-hour traffic in cities
  - Telephone call centers
  - Seasonal demands for a manufactured product
- **It can be critical to model this nonstationarity for model validity**
  - Ignoring peaks, valleys can mask important behavior
  - Can miss rush hours, etc.
- **Good model: *Nonstationary Poisson process***



Nonstationary Arrival Processes (cont'd.)

- **Two issues:**
  - How to specify/estimate the *rate function*
  - How to generate from it properly during the simulation (will be discussed in Chapters 8, 11 …)
- **Several ways to estimate rate function — we'll just do the *piecewise-constant* method**
  - Divide time frame of simulation into subintervals of time over which you think rate is fairly flat
  - Compute observed rate within each subinterval
  - Be *very* careful about time units!
    - Model time units = minutes
    - Subintervals = half hour (= 30 minutes)
    - 45 arrivals in the half hour; rate = 45/30 = 1.5 *per minute*



Multivariate and Correlated Input Data

- **Usually we assume that all generated random observations across a simulation are independent (though from possibly different distributions)**
- **Sometimes this isn't true:**
  - A "difficult" part requires long processing in both the Prep and Sealer operations
  - This is positive correlation
- **Ignoring such relations can invalidate model**
- **See textbook for ideas, references**

Chapter 6
Intermediate Modeling and Terminating Statistical Analysis

By : Arya Wirabhuana

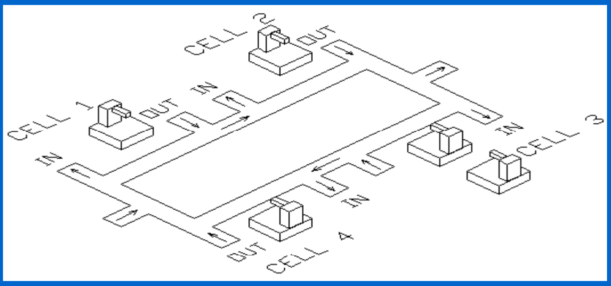
Department Of Industrial Engineering

UIN SUNAN KALIJAGA YOGYAKARTA

What We’ll Do ...

- **A small manufacturing system (Model 6.1)**
  - Sequences
  - Variables and Expressions
  - Sets
  - Importing CAD drawings
  - Verification (a.k.a. debugging)
- **Statistical analysis of output data: terminating simulations**
  - Time frame of simulations
  - Replicating and confidence intervals (Model 6.2)
  - Comparing alternatives

A Small Manufacturing System



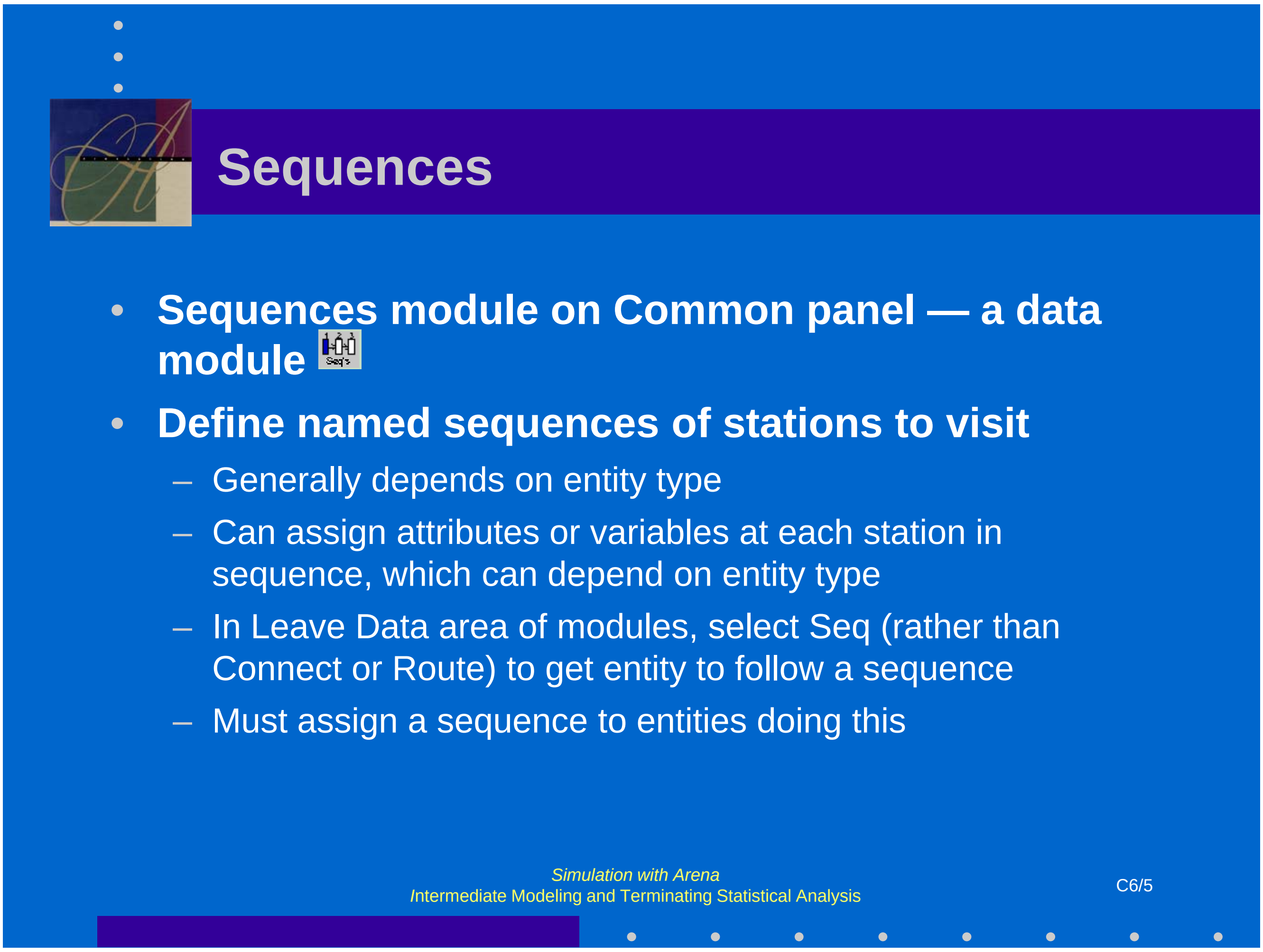
- **Part arrivals, four cells, part departures**
- **Cells 1, 2, and 4: single machine each**
- **Cell 3: two machines — newer one 20% faster**
  - Need: way to model non-identical resource units
- **Circular layout of cells**
- **Parts enter at left, exit at right, travel only clockwise, all transfer times = 2 min.**

A Small Manufacturing System (cont'd.)

- **Three separate part types**
  - Interarrivals (all types merged) ~ expo(13) minutes
  - 26% type 1, 48% type 2, 26% type 3
- **Different part types follow different routes, have different (triangular) processing times:**

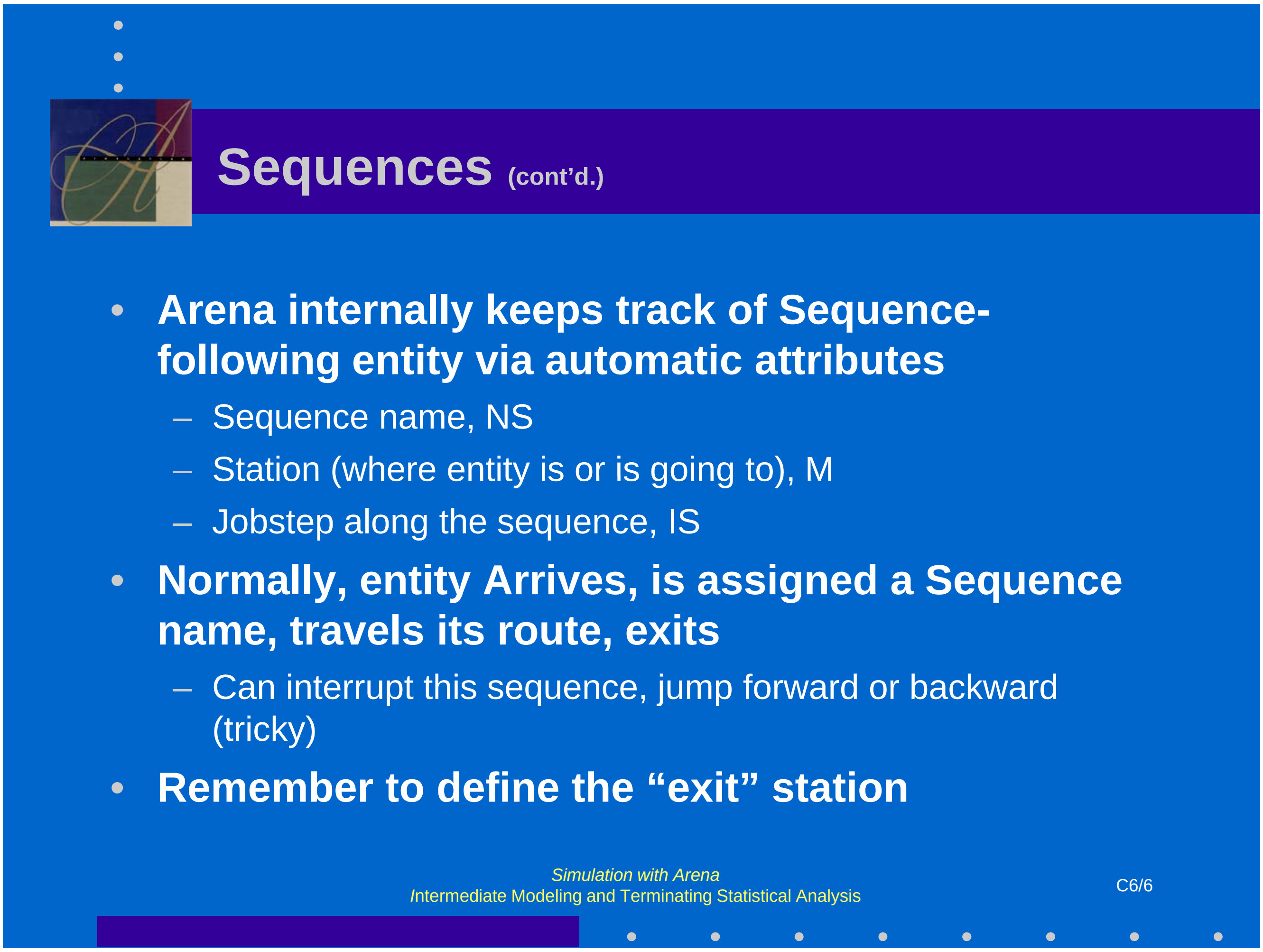
Part Type	Cell/ Time	Cell/ Time	Cell/ Time	Cell/ Time	Cell/ Time
1	**1** **6, 8, 10**	**2** **5, 8, 10**	**3** **15, 20, 25**	**4** **8, 12, 16**	
2	**1** **11, 13, 15**	**2** **4, 6, 8**	**4** **15, 18, 21**	**2** **6, 9, 12**	**3** **27, 33, 39**
3	**2** **7, 9, 11**	**1** **7, 10, 13**	**3** **18, 23, 28**		

  - Need: way to change routing depending on part type — *process plans*



Sequences

- **Sequences module on Common panel — a data module**
- **Define named sequences of stations to visit**
  - Generally depends on entity type
  - Can assign attributes or variables at each station in sequence, which can depend on entity type
  - In Leave Data area of modules, select Seq (rather than Connect or Route) to get entity to follow a sequence
  - Must assign a sequence to entities doing this



Sequences (cont'd.)

- **Arena internally keeps track of Sequence-following entity via automatic attributes**
  - Sequence name, NS
  - Station (where entity is or is going to), M
  - Jobstep along the sequence, IS
- **Normally, entity Arrives, is assigned a Sequence name, travels its route, exits**
  - Can interrupt this sequence, jump forward or backward (tricky)
- **Remember to define the "exit" station**

Variables

- **Allow re-use of the same number(s) in different places**
- **Can only be constant values, but any entity can reassign the value of a Variable**
- **Variables module from Common panel** X,Y Var'bles
  - Data module
  - Defines names, initial values of Variables
  - Can be a scalar, vector, or 2-dim. matrix
- **Cannot involve arithmetic, entity attributes, other Variables, or distributions**

Expressions

- **Similar motivation to Variables — re-use the same "thing" in several places in the model**
- **A fixed "formula" or function that *can* involve arithmetic, entity attributes, other Variables, and distributions — very general**
- **However, the *form* of the expression cannot be changed during the simulation**
- **Expressions module from Common panel** y=x Expr's
  - Data module
  - Defines names, form of Expressions
  - Can be a scalar, vector, or 2-dim. matrix

Sets

- Group similar objects (resources, sequences, pictures, other) together under a single name
- Define: Sets module from Common panel {abc} Sets
- Refer to objects in a Set by their original name (independent of the Set membership) or by an index into the Set
- An object can be a member of more than one Set (or not be in any Sets)
- Can form Sets of just about anything
  - A given Set must have the same kind of members

Sets (cont'd.)

- Perhaps the most common Sets: Resources
  - Allows dissimilar resources to be grouped — more general than multi-Capacity single Resources, where they all have to be identical
  - Entities can choose among members of a Resource Set according to preferences, rules
  - Can animate individual Resources in Set (state, picture) — unlike multi-Capacity single Resources
- Also group Sequences and entity pictures into sets for ease of access (via part-type number)

Modeling Approach

- **There are usually several different ways to represent a model in software**
- **Often driven by data requirements, availability**
- **Decide on and write down ahead of time how things will be represented; here:**
  - Sequences (in a Set) to control part flow
  - Assign entities in Sequences for process times except for Cell 1
  - Use Expression vector for process times at Cell 1
  - Use Variables for transfer times, speed factor at Cell 3

Building the Model

- **Place the data modules (Sequences, Expressions, Variables, Sets, Simulate)**
  - Defines names, initial values
  - Makes names available on pull-down lists for reference later while editing logic modules
- **Place the logic modules, making use of names previously defined in data modules**
- **Animation — will import AutoCAD drawing**
- **Verification of expression of model in software**
- **Statistical design and analysis**

Sequences Module

- **Define station paths for different part types**
- **Name the sequences**
  - `Part 1 Process Plan`, etc.
- **Can define actions at each station as part of the definition of the sequences**
  - Define `Process Time` attribute (except at Cell 1)
- **Group the sequences into a Set (later)**
- **Assign a sequence, indexed by set, to each arriving part once its type is generated**
- **Don’t forget to define “exit” for each sequence**
- **Good (but not the only) way to model this**

Expressions Module

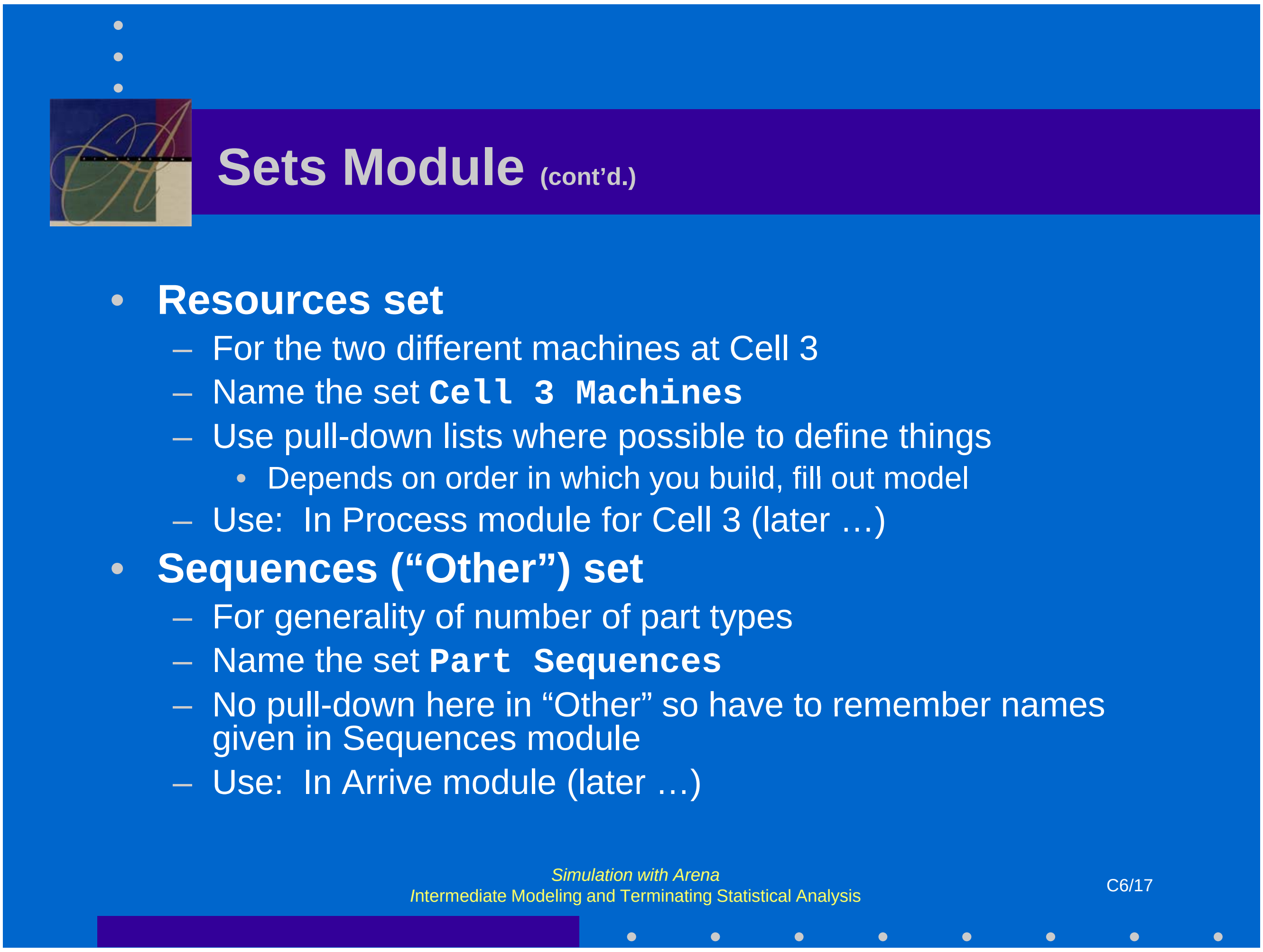
- **Many potential uses, but here to define process times at Cell 1 for different part types**
- **Name each expression: `Cell 1 Times`**
- **Scalar, vector, or 2-dimensional array**
  - Use 3x1 vector here, one for each part type
  - Must index in correct order — here, by part type
- **Can be very general expression**
  - Here, process-time distributions at Cell 1
  - Entities will index into vector by their part type: `Cell 1 Times(Part Index)`
- **Again, this is not the only way to model this**

Variables Module

- **Define global variables, their initial values**
  - Like Expressions, these are global to the model
  - Unlike Expressions, they can only take on numerical values, not formulas or other variables, expressions, or attributes (but the numerical value can be changed during the run by any entity)
  - Here, use for two different variables
  - Speed factor at Cell 3: 2x1 vector **`Factor`**
    - 0.8 for index 1 (new machine), 1.0 for index 2 (old)
  - Value (2) of part transfer times: **`Transfer Time`**
- **(Yet) again, this is not the only way to do this**

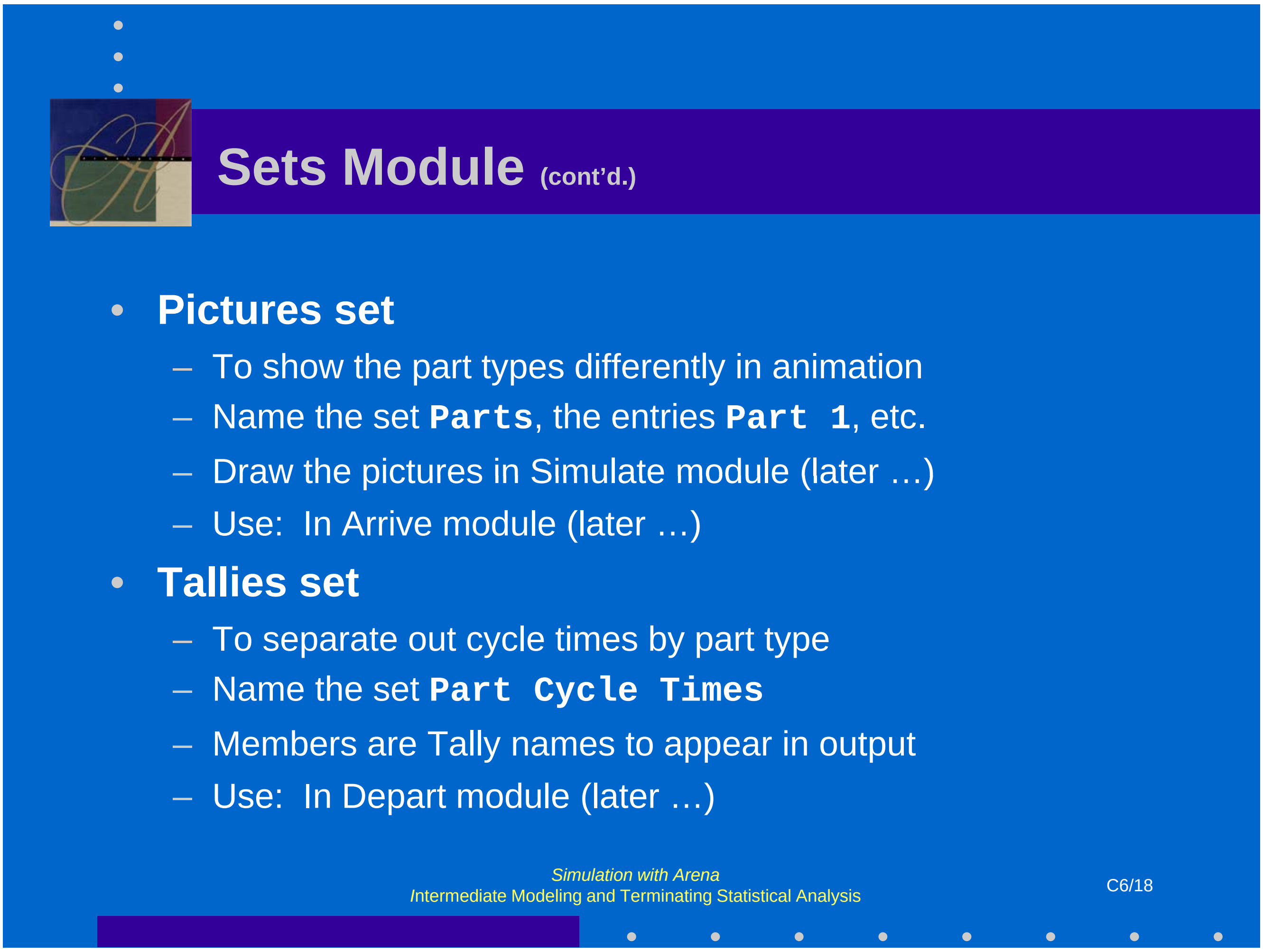
Sets Module

- **Group together (and name and index) “like” things — Resources, Queues, Storages, Stations, Pictures, Counters, Tallies, and “Other” (e.g., Sequences)**
- **Can make models more general, “cleaner”**
  - In some cases, needed to model things properly, like workers (Resources) with different but overlapping skills
- **Order of entry of set members is critical!**
- **Here, we’ll form sets of Resources, Sequences (“Other”), Pictures, and Tallies**



Sets Module (cont'd.)

- **Resources set**
  - For the two different machines at Cell 3
  - Name the set **`Cell 3 Machines`**
  - Use pull-down lists where possible to define things
    - Depends on order in which you build, fill out model
  - Use: In Process module for Cell 3 (later …)
- **Sequences ("Other") set**
  - For generality of number of part types
  - Name the set **`Part Sequences`**
  - No pull-down here in "Other" so have to remember names given in Sequences module
  - Use: In Arrive module (later …)



Sets Module (cont'd.)

- **Pictures set**
  - To show the part types differently in animation
  - Name the set **`Parts`**, the entries **`Part 1`**, etc.
  - Draw the pictures in Simulate module (later …)
  - Use: In Arrive module (later …)
- **Tallies set**
  - To separate out cycle times by part type
  - Name the set **`Part Cycle Times`**
  - Members are Tally names to appear in output
  - Use: In Depart module (later …)

Simulate Module

- **Identify model and define Length of Replication, as in earlier models**
- **Also, draw the three entity Pictures and associate with names of members of Pictures set**
  - Double-click on “generic” entity picture above module handle
  - Add new or edit existing pictures in list
    - Copy current last entry to add to bottom of list
    - Draw: shapes, size; colors for fill, line, text, etc.
    - Close drawing window to accept
  - Names (Values) must already be assigned in Sets

Arrive Module

- **Main dialog**
  - Station name: **`Order Release`**
  - Time Between: **`EXPO(13)`**
  - Mark Time Attribute: **`Enter Time`**
  - Leave Data
    - Route: *select* (this is the default anyway)
    - Seq: *select*
    - Route Time: **`Transfer Time`**

Arrive Module (cont'd.)

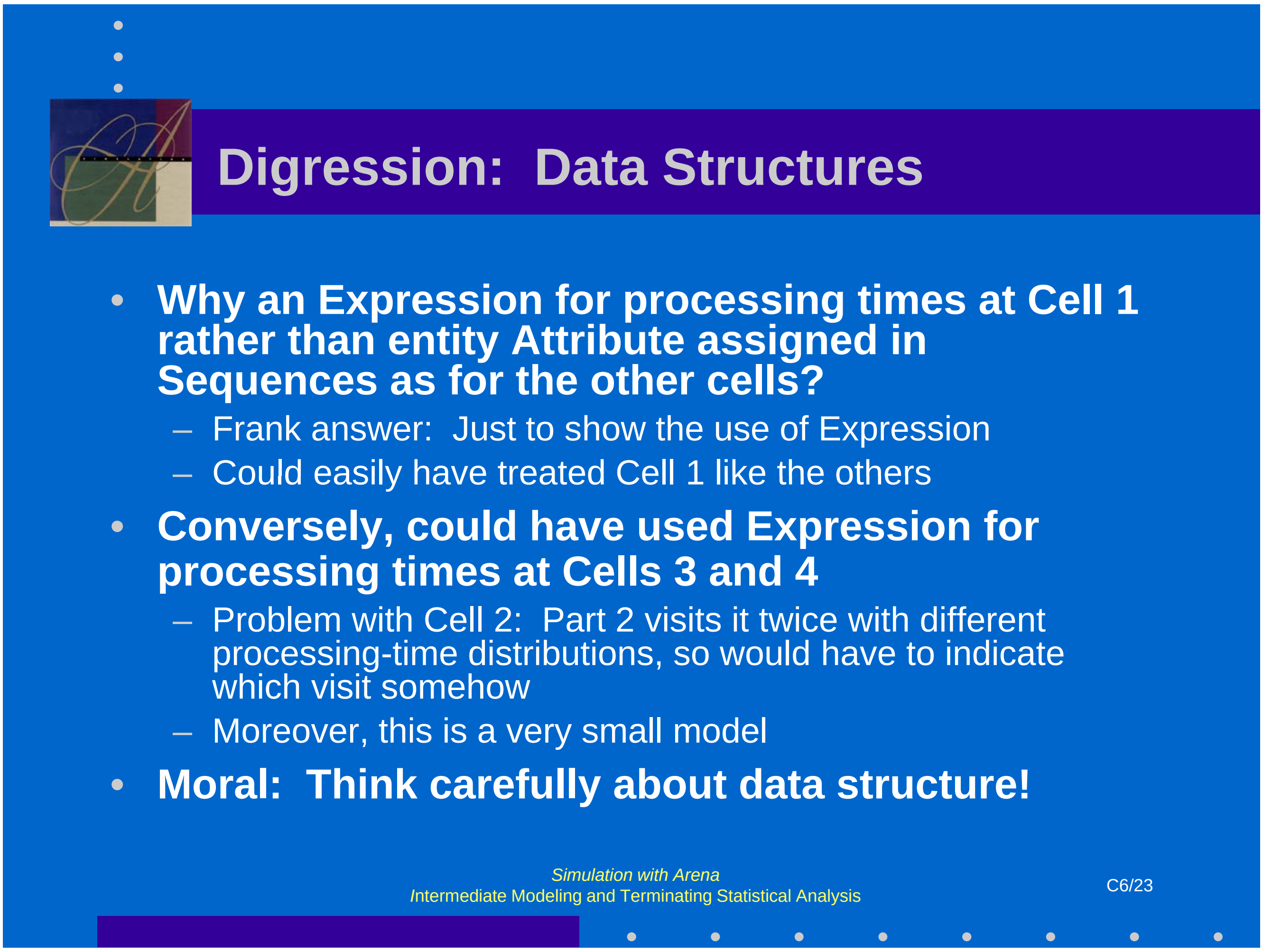
- **Assign… subdialog**
  - Attribute: **`Part Index = DISC(…)`**(*cumulatives*)
  - Other: Sequence = **`Sequence Set(Part Index)`**

Was just defined

- **Animate… subdialog**
  - Initial Entity Picture
    - Set Member: *select*
    - Picture Set: **`Parts`**
    - Set Index: **`Part Index`**

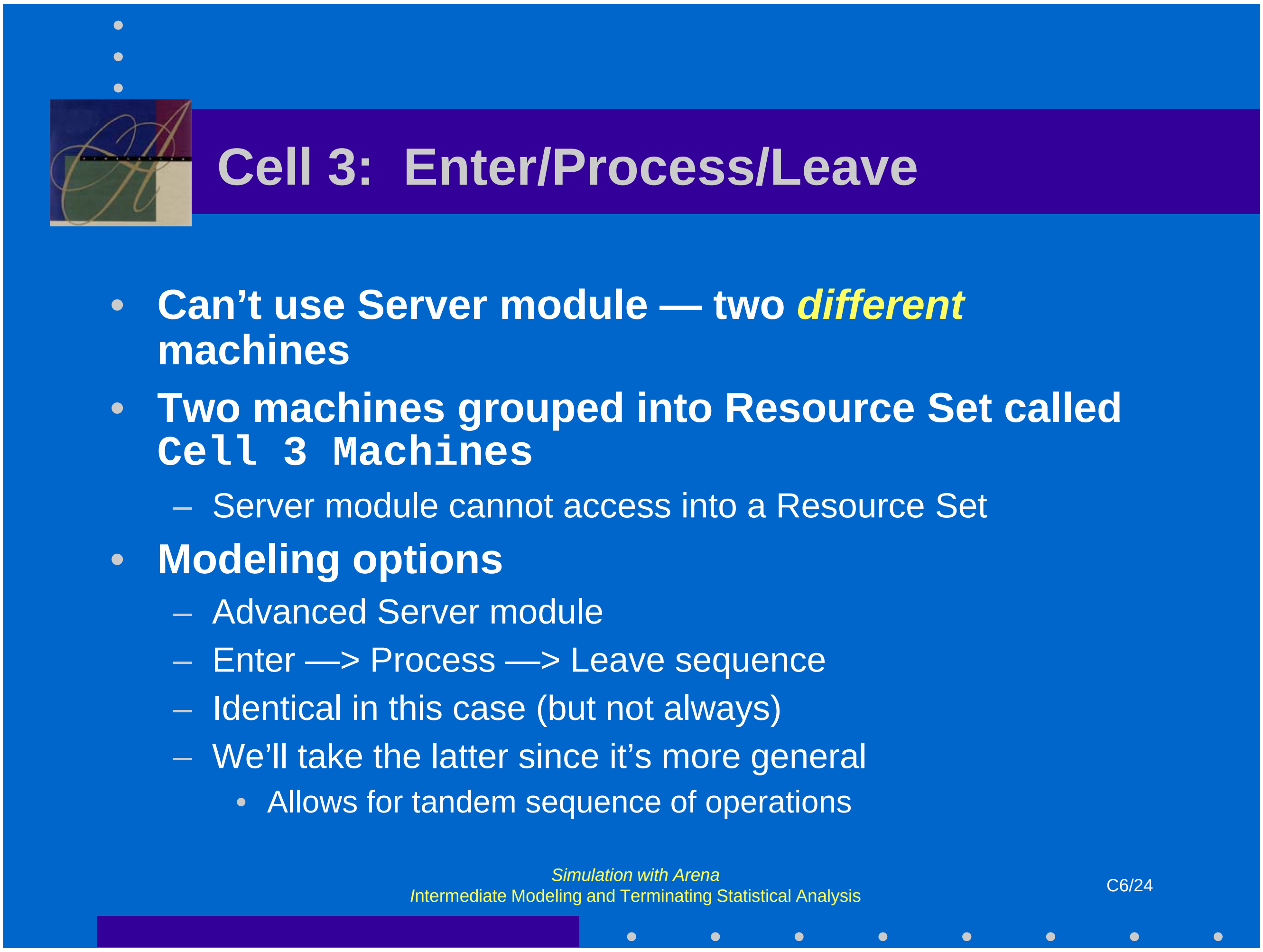
Server Modules for Cells 1, 2, 4

- **Main dialog**
  - Station: **`Cell 1`** (or **`Cell 2`** or **`Cell 4`**)
  - Process Time:
    - If Cell 1: **`Cell 1 Times(Part Index)`**
      - Global Expression defined in Expressions module
    - If Cell 2 or 4: **`Process Time`**
      - Entity Attribute defined as part of Sequences
  - Leave Data:
    - Route: *select*
    - Seq: *select*
    - Route Time: **`Transfer Time`**
      - Global Variable defined in Variables module



Digression: Data Structures

- **Why an Expression for processing times at Cell 1 rather than entity Attribute assigned in Sequences as for the other cells?**
  - Frank answer: Just to show the use of Expression
  - Could easily have treated Cell 1 like the others
- **Conversely, could have used Expression for processing times at Cells 3 and 4**
  - Problem with Cell 2: Part 2 visits it twice with different processing-time distributions, so would have to indicate which visit somehow
  - Moreover, this is a very small model
- **Moral: Think carefully about data structure!**



Cell 3: Enter/Process/Leave

- **Can't use Server module — two *different* machines**
- **Two machines grouped into Resource Set called `Cell 3 Machines`**
  - Server module cannot access into a Resource Set
- **Modeling options**
  - Advanced Server module
  - Enter —> Process —> Leave sequence
  - Identical in this case (but not always)
  - We'll take the latter since it's more general
    - Allows for tandem sequence of operations

Cell 3: Enter/Process/Leave (cont'd.)

- **Enter module: generic "landing pad" for entities**
  - Station: *select*
  - Station: `Cell 3`
  - Queue: *Storage* for unload time (can delete here)
- **Process module: describe processing delay**
  - More general than Server Data area of Server module — can reference into a Resource Set
  - Activity is to *Seize* a member of a Resource Set
  - Resource Set: `Cell 3 Machines`
  - Rule: Cyclical (cycle around available members)
  - Store Index in Att: `Index` (no. of chosen member)
  - Process Time: `Process Time * Factor(Index)`

0.8 for Index = 1 (new)
1.0 for Index = 2 (old)

Cell 3: Enter/Process/Leave (cont'd.)

- **Leave module: generic "launching pad"**
  - Must specify Station from which entity is leaving
  - From Station: `Cell 3`
  - Seq: *check*
  - Route Time: `Transfer Time`
- **Must Connect Enter - Process - Leave**
  - Could also use Labels but wouldn't be visible
- **Due to its position in the layout, want to re-orient these modules for right-to-left flow**
  - Arrange the modules physically that way
  - Move entry (box) and exit (triangle) points
    - Flip triangle about vertical line to reorient

Resource Definitions

- **Must define Cell 3 machine resources explicitly**
- **Resource modules**
  - Separate ones for old, new machine
  - Need only define Name, Type, Capacity for each
    - Names: **`Cell 3 New`**, **`Cell 3 Old`**
    - Both are Capacity Type with Capacity = 1
  - Both have animation picture, but won’t be used (will use Enter module station) so can be deleted
  - Both have Individual Queue, but won’t be used (will use queue on Process module) so can be deleted

Parts’ Exit

- **Use Depart module, as before**
- **Need to collect cycle times by part type**
  - Station name: **`Exit System`**
  - Tally Set Member: *select*
  - Tally Set: **`Part Cycle Time`**
  - Set Index: **`Part Index`**
  - Type of Statistics: Interval (*select*)
    - Attribute: **`Enter Time`**
  - Will produce three different Tally reports, one for each part type

Animation

- **Could run model at this point, would get correct numerical summary results**
  - But part movement would not show in animation
- **Pull animation away from logic, data modules**
- **Move, resize, reorient queues for realism**
- **Animate Routes (all movement possibilities)**
  - Thick “bundles” of routes — Shift key, Snap to Grid
  - Heed clockwise direction
  - Draw lines to define route “lanes”
- **Import AutoCAD dxf file for backdrop (see text)**
- **Fine-tune resource pictures**
  - Layers for seize point

Verification

- **System —> Model —> “Code”**
- ***Validation*: Is Model = System?**
- ***Verification*: Is “Code” = Model? (debugging)**
- **The Truth: Can probably never completely verify, especially for large models**

Verification (cont'd.)

- **Some techniques to *attempt* verification**
  - Eliminate error messages (obviously)
  - Single entity release, Step through logic
    - Set Max Batches = 1 in Arrive
    - Replace part-type distribution with a constant
  - "Stress" model under extreme conditions
  - Performance estimation — like slide-rule decimal placement
  - Look at generated SIMAN .mod and .exp files
    - *Run/SIMAN/View* menu option

Statistical Analysis of Output Data: Terminating Simulations

- **Random input leads to random output (RIRO)**
- **Run a simulation (once) — what does it mean?**
  - Was this run "typical" or not?
  - Variability from run to run (of the same model)?
- **Need statistical analysis of output data**
- **Time frame of simulations**
  - Terminating: Specific starting, stopping conditions
  - Steady-state: Long-run (technically forever)
  - Here: Terminating

Modify Model 6.1 into Model 6.2

- **Establish a realistic termination rule**
  - There are many different ways to terminate
  - Really a modeling issue — what's realistic?
- **Process an incoming order of 100 parts**
  - Set Max Batches = 100 in Arrive
  - Shuts off Arrival stream after 100 entities created
  - *In this model*, will cause termination when 100th part exits since there are no other events and the event calendar will become empty
  - Causes system to become less congested when the end of the simulation is approached

Modify Model 6.1 into Model 6.2 (cont'd.)

- **Establish a single overall performance measure: *Work in Process* (WIP)**
  - Variable `WIP` defined (Variables module)
  - Click `WIP` up 1 just after each arrival
    - Arrive module, Assign Variable `WIP = WIP + 1`
  - Click `WIP` down 1 just before each departure
    - No Assign subdialog in Depart module
    - Put *Actions* module just before Depart to click down
    - Re-wire Sequences, Routes to send entities here instead of to `Exit System` Depart module
    - Direct Connect to `Exit System` Depart module
  - Add Statistics module to request collection and reporting of Time-Persistent stats on `WIP`

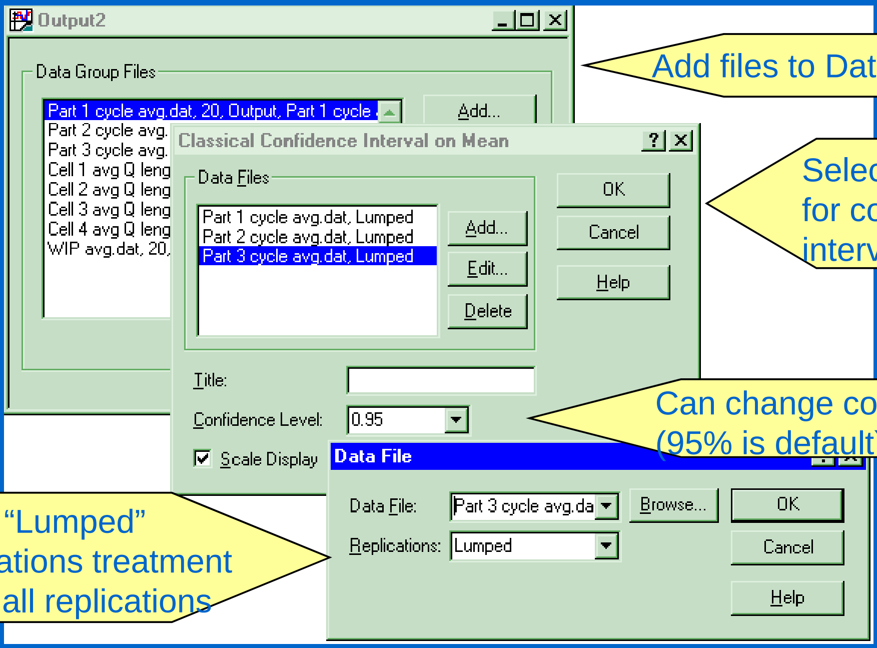
Strategy for Data Collection and Analysis

- **For terminating case, make IID replications**
  - Simulate module: Number of Replications field
  - Check both boxes for Initialization Between Reps.
  - Get multiple independent Summary Reports
- **How many replications?**
  - Trial and error (now)
  - Approximate no. for acceptable precision (below)
  - Sequential sampling (Chapter 11)
- **Save summary statistics across replications**
  - Statistics Module, Outputs Area, save to files
- **Maybe turn off animation (*Run/Setup/Mode*)**

Confidence Intervals for Terminating Systems

- **Output Analyzer on files saved from Outputs area (cross-replication) of Statistics module**
- **Define, read in, save Data Group(s)**
- **In Output Analyzer**
  - *Analyze/Conf. Interval on Mean/Classical…* menu (or +)
  - Add desired files; select Lumped for Replications

Confidence Interval Dialogs



Output2 — Data Group Files:

- Part 1 cycle avg.dat, 20, Output, Part 1 cycle
- Part 2 cycle avg.
- Part 3 cycle avg.
- Cell 1 avg Q leng
- Cell 2 avg Q leng
- Cell 3 avg Q leng
- Cell 4 avg Q leng
- WIP avg.dat, 20,

Classical Confidence Interval on Mean — Data Files:

- Part 1 cycle avg.dat, Lumped
- Part 2 cycle avg.dat, Lumped
- Part 3 cycle avg.dat, Lumped

Title:

Confidence Level: 0.95

☑ Scale Display

Data File: Part 3 cycle avg.da

Replications: Lumped

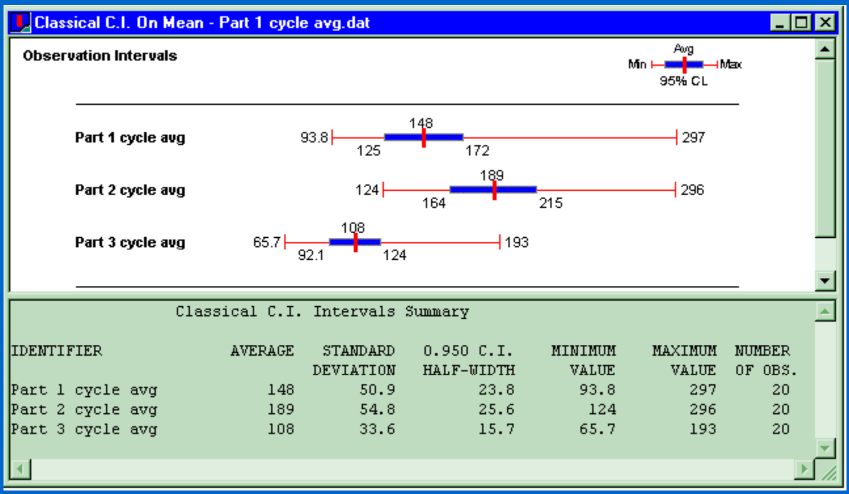
Add files to Data Group

Select files for confidence intervals

Can change confidence level (95% is default)

Select “Lumped” Replications treatment to use all replications

Confidence Interval Results



```
                         Classical C.I. Intervals Summary

IDENTIFIER                AVERAGE    STANDARD   0.950 C.I.    MINIMUM    MAXIMUM   NUMBER
                                     DEVIATION   HALF-WIDTH     VALUE      VALUE    OF OBS.
Part 1 cycle avg              148       50.9        23.8        93.8        297       20
Part 2 cycle avg              189       54.8        25.6         124        296       20
Part 3 cycle avg              108       33.6        15.7        65.7        193       20
```

- **Interpretation of confidence intervals**
  - What’s being estimated
  - Coverage, precision, robustness to non-normality

Automatic Text-Only 95% Confidence Intervals

- **At end of summary report, get 95% confidence intervals as above, in text-only format, *if***
  - You ask for more than one replication, *and*
  - You have a Statistics module with Outputs entries
- **Done only for output measures in Statistics module's Outputs area**

```
                                             OUTPUTS
Identifier                 Average     Half-width Minimum     Maximum # Replications
________________________________________________________________________________________
avg WIP                    11.327      1.7801      6.3666      18.704      20
Part 1 cycle avg           148.38      23.809      93.763      296.79      20
Part 2 cycle avg           189.47      25.602      124.38      296.29      20
Cell 4 avg Q length        1.5652      .64270      .48813      6.4610      20
Cell 2 avg Q length        1.7943      .66541      .45236      6.3355      20
Cell 1 avg Q length        1.7960      .52429      .45934      4.5093      20
Part 3 cycle avg           107.80      15.704      65.692      193.10      20
Cell 3 avg Q length        1.2976      .40001      .45141      3.3876      20
```

Half Width and Number of Replications

- **Prefer smaller confidence intervals — *precision***
- **Notation:** $n =$ no. replications
  $\bar{X} =$ sample mean
  $s =$ sample standard deviation
  $t_{n-1,1-\alpha/2} =$ critical value from t tables
- **Confidence interval:** $\bar{X} \pm t_{n-1,1-\alpha/2} \frac{s}{\sqrt{n}}$
- **Half-width =** $t_{n-1,1-\alpha/2} \frac{s}{\sqrt{n}}$

Want this to be "small," say $\le h$ where h is prespecified

- **Can't control *t* or *s***
- **Must increase *n* — how much?**

Half Width and Number of Replications (cont'd.)

- **Set half-width = *h*, solve for** $n = t^2_{n-1,1-\alpha/2}\frac{s^2}{h^2}$
- **Not really solved for *n* (*t*, *s* depend on *n*)**
- **Approximation:**
  - Replace *t* by *z*, corresponding normal critical value
  - Pretend that current *s* will hold for larger samples
  - Get $\boxed{n \cong z^2_{1-\alpha/2}\frac{s^2}{h^2}}$ *s* = sample standard deviation from "initial" number n_0 of replications
- **Easier but different approximation:**

$$\boxed{n \cong n_0 \frac{h_0^2}{h^2}}$$

h_0 = half width from "initial" number n_0 of replications

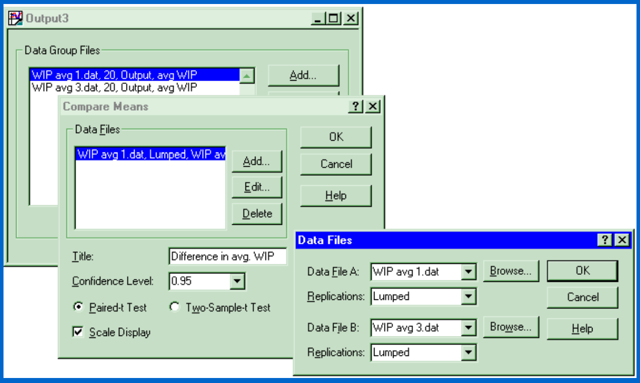
n grows quadratically as *h* decreases.

Comparing Alternatives

- **Usually, want to compare alternative system configurations, layouts, scenarios, sensitivity analysis …**
- **Here: Transfer time (2 min) smells like a guess — does it matter if it's, say, 1 vs. 3?**
  - Call these alternatives A and B in Arena
- **Single measure of performance: average WIP**
- **Make two sets of 20 replications, for A and B**
  - Must rename output files to distinguish them

Comparing Alternatives (cont'd.)

- ***Analyze/Compare Means*** **menu (no button)**
  - Add comparable data files for A and B
  - Lumped Replications



Output3

Data Group Files

- WIP avg 1.dat, 20, Output, avg WIP
- WIP avg 3.dat, 20, Output, avg WIP

Compare Means

Data Files: WIP avg 1.dat, Lumped, WIP av

Title: Difference in avg. WIP

Confidence Level: 0.95

● Paired-t Test ○ Two-Sample-t Test

☑ Scale Display

Data Files

Data File A: WIP avg 1.dat

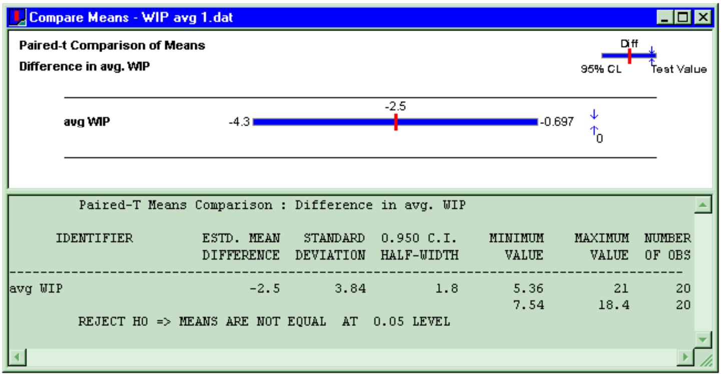
Replications: Lumped

Data File B: WIP avg 3.dat

Replications: Lumped

Comparing Alternatives (cont'd.)

- **Results:**



Compare Means - WIP avg 1.dat

Paired-t Comparison of Means

Difference in avg. WIP

```
          Paired-T Means Comparison : Difference in avg. WIP

      IDENTIFIER         ESTD. MEAN  STANDARD  0.950 C.I.    MINIMUM    MAXIMUM  NUMBER
                         DIFFERENCE  DEVIATION  HALF-WIDTH     VALUE      VALUE  OF OBS
-----------------------------------------------------------------------------------------
avg WIP                       -2.5       3.84         1.8       5.36         21      20
                                                                7.54       18.4      20
          REJECT H0 => MEANS ARE NOT EQUAL  AT  0.05 LEVEL
```

- **c.i. on difference misses 0, so conclude that there *is* a (statistically) significant difference**

Chapter 7
Entity Transfer and Steady-State Statistical Analysis

By : Arya Wirabhuana

Department Of Industrial Engineering

UIN SUNAN KALIJAGA YOGYAKARTA

What We'll Do ...

- **Types of Entity Transfers**
- **Resource-Constrained Transfers**
- **Transporters (Model 7.1)**
- **Conveyors**
  - Non-accumulating (Model 7.2)
  - Accumulating (Model 7.3)
- **Statistical analysis of steady-state simulations**
  - Warm up and run length (Model 7.4)
  - Truncated replications (Model 7.5)
  - Batching in a single run (Model 7.6)
  - Automatic run-time confidence intervals via batch means

Types of Entity Transfers

- **Connect**
  - Zero-delay
  - Connection graphic vs. module Labels (no graphic)
- **Route**
  - Non-zero-delay — constant, r.v., expression
  - Stations, animated Routes
  - Fixed routes vs. entity-dependent Sequences
- **Connect and Route both assume:**
  - No limit on number in transit at a time
  - Entities have their own feet

Resource-Constrained Transfers

- **Limit total number of entities in transit at a time**
  - Telecommunications (number of packets)
  - Logistics (number of vehicles)
- **Entities still have their own feet**
- **Model via existing constructs — think creatively**
  - Model “space” on the “road” as a Resource
  - Limit number of Units of this Resource
  - Entity must Seize unit of “space” resource before beginning trip, Release it at end of trip
- **The “space” resource must be defined via a Resources module if > 1 unit is required**

Resource-Constrained Transfers (cont'd.)

- **Leave Data area of many logic modules**
  - Tran Out… button
  - Seize a unit of the "space" resource
  - Priority (1 is highest, …)
  - If no units are free, entity must stay where it is
  - Option for Load Time
- **Enter Data area of many logic modules**
  - Tran In… button
  - Release the unit of the "space" resource
  - Option for Unload Time

Transporter Concepts

- **Carts, fork lifts, trucks, wheelchairs, people, …**
- **When entity is ready to go somewhere, it needs to be "picked up" and moved**
- **Use *Transporters* — "moveable" resources**
- **Activities: Request, Transport, and Free**
- **Two types of Transporters**
  - *Free-Path* (we'll do)
    - Travel time depends only on velocity, distance
    - Ignore "traffic jams" and their resulting delays
  - *Guided* (won't do)
    - AGVs, intersections, etc.

Creating Transporters

- **Transporter data module (*Transfer panel*)**
  - Transporter name (**`Cart`** in our model)
  - Number of Units (**2**)
  - Free Path vs. Guided (accept Free Path default)
  - Distance Set (accept default name **`Cart_Dst`**)
  - Velocity (**`50`** feet/min.)

    Consistency of time and distance units across model
  - Initial Position (accept Arbitrary default)
  - Options ... When Freed ... (Remain Where Freed)
  - Can have Schedules, Failures, Statistics
  - Picture for animation, including Ride Point

Requesting a Transporter

- **Entity is done at a station, wants to move on**
- **Leave Data area, Tran Out… button**
  - Transfer Type: Request
  - Transporter name (**`Cart`**)
  - Rule (**`Smallest Distance`**)
    - Applies when > 1 transporter is available
    - Other options include Cyclical, Preferred Order, Random
  - Priority of this request (**`1`**)
    - Applies when demand for transporters exceeds supply
  - Load Time (**`0.25`** min.)

Entity Animation During the Transporter Request Process

- **Use an Arena *Storage* (similar to a queue)**
  - Place to store entity to make it show in animation
  - Get at via animate subdialog in most logic modules
    - Leave for Next Station area
    - Select Storage radio button
    - Storage name (accept default)
    - Check Unstore box to decrement Storage statistics
  - Produces animation object that looks like an animated queue, except for default color (green)
    - Can edit the storage animation just like a queue

Freeing a Transporter

- **Entity has arrived at its next station on a Transporter**
- **Enter Data area of this station, Tran In… button**
  - Select Free Transporter radio button
    - Arena keeps track of which one, unit number
  - Unload Time (`0.25` min.)
- **Transporter is now available, will position itself according to Options ... When Freed … in Transporter data module**

Distances for Transporters

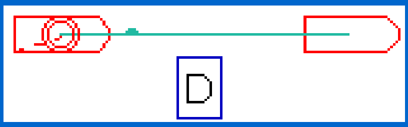
- **Define contents of the Distance Set `Cart_Dst`**
- **Distances (in feet) moved by parts:**

		To				
		Cell 1	Cell 2	Cell 3	Cell 4	Exit System
F	Order Release	37	74			
r	Cell 1		45	92		
o	Cell 2	139		55	147	
m	Cell 3				45	155
	Cell 4		92			118

Units!!

  - Blank cells: part movements that don't occur
- **Place 11 Distance modules (Transfer panel)**
  - Beginning, Ending Stations
  - Member of what Distance Set?
  - Numerical Distance Units!!
  - Direction is implied — need not be symmetric

Animating Transporter Movement



- **Part of Distance module**
- **Beginning, Ending stations**
  - First, move module handle to appropriate physical location (organize Distance modules — maybe many)
  - Then, drag animation (stations, line) where they belong
    - Dragging module handle also moves animation
    - Use Shift+Drag on handle to leave animation where it is

Animating Transporter Movement (cont'd.)

- **Line characteristics — double-click on line**
  - # Points — then drag to correct animation position
  - Maybe Rotate/Flip entity as it moves, line Colors
- **Parking Areas for Transporters**
  - Double-click on double circle, Points…, Add, …
  - Like a queue — Points vs. Line, Color, Rotate, Flip
    - Drag around for realistic positioning
  - There could be many animation stations for a single logical station, but you should have only one parking area per Station (delete others)

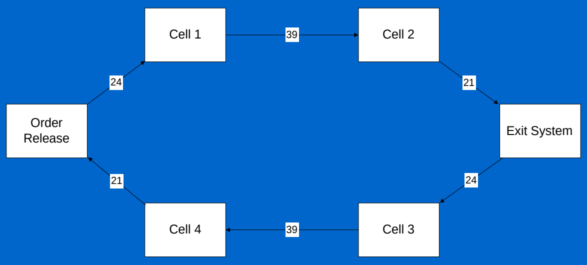
More Distances — Empty Transporters

- **Above Distances incomplete — only for entities**
- **Transporters must also move when empty**
  - In general, $n(n - 1)$ distances need definition for network with n nodes
  - Some not possible — Order Release to Exit System
- **Additional distances to define (grayed):**

		To					
		Order Release	Cell 1	Cell 2	Cell 3	Cell 4	Exit System
	Order Release		37	74			
F	Cell 1	155		45	92	129	
r	Cell 2	118	139		55	147	
o	Cell 3	71	92	129		45	155
m	Cell 4	34	55	92	139		118
	Exit System	100	121	158	37	74	

System Improvement (?): Replace Transporters with Conveyor

- Replace Transporters with a conveyor
- Loop conveyor to follow main path, clockwise
- Six entrance/exit points
  - Load, Unload takes 0.25 minute
  - Each part is 4 feet per side, but want 6 feet of conveyor space for clearance on corners
- Speed = 20 feet/minute Units!!
- Distances:



Order Release → 24 → Cell 1

Cell 1 → 39 → Cell 2

Cell 2 → 21 → Exit System

Exit System → 24 → Cell 3

Cell 3 → 39 → Cell 4

Cell 4 → 21 → Order Release

Conveyor Concepts

- Entity to be conveyed must wait for space
- Conveyor consists of ***cells***
  - Equal size, constantly moving
- Entities might require multiple contiguous cells
- Must define ***cell size***; tradeoff involved:
  - Small cells: accurate model but slow execution
  - Large cells: just the opposite!
- Entities ***Access*** space, ***Convey***, and ***Exit***
- Conveyor = series of linear ***Segments***
  - Each segment starts and ends with a Station
  - Link to form loops, diverge points, converge points

Types of Conveyors

- **Both travel in a single, irreversible direction**
- ***Nonaccumulating*: belt, bucket line, escalator**
  - Spacing between entities on it doesn't change
  - Entire conveyor stops for entity Access/Exit if Load/Unload time is > 0
- ***Accumulating*: rollers, freeway**
  - Conveyor never stops moving
  - If entity on it stops to Exit, other entities behind it are blocked and bunch up (entities ahead of it keep moving)
  - When blockage ends, blocked entities go on but maybe not all at once (spacing requirements)

Change Transporters to Non-Accum. Conveyors (Model 7.2)

- **Modify Transporter model (Model 7.1)**
- **Define Variables `Load Time`, `Unload Time`**
- **Delete Transporter module**
- **Add Conveyor module (Transfer panel)** Convyr
  - Name, Segment Set (define Segments later)
  - Velocity, Cell Size definitions
  - Initial status: Active
  - Max Cells Occupied (for "biggest" possible entity)
  - Type: Nonaccumulating
  - Entity Size: Cell Size
  - Conveyor statistics (check box)

Change Transporters to Non-Accum. Conveyors (Model 7.2) (cont'd.)

- **Every Tran Out: change Request to Access**
  - Conveyor = **`Loop Conveyor`**
  - # of Cells to Access = **2**
  - Load Time = **`Load Time`** (Variable name)
- **Every Tran In: change Free Transporter to Exit Conveyor**
  - Unload Time = **`Unload Time`** (Variable name)
- **Replace Distances with Segments (need 6)**
  - Beginning, Ending Stations, Segment Set Name
  - Define distances for simulation
    - Units = actual length (in feet), not number of cells
  - Place lines for animation

Digression: Manual Change of Variable Values During Run

- **Increase `Load Time` and `Unload Time` to 2 *during* run (see effect on conveyor stopping)**
- **Pause the run:** ⏸
  - Suppose you do this at time TNOW = 301.885
- **Use Run Controller: *Run/Command* menu option or**

```
301.885>ASSIGN Load Time = 2
301.885>ASSIGN Unload Time = 2
```

- **Resume the run: Close Controller window;** ▶
- **OK for demos; questionable for "real" runs**

Change Conveyors to Accumulating Type (Model 7.3)

- **Modify Model 7.2 (with non-accumulating conveyors)**
- **Conveyor module**
  - Change Type to Accumulating
  - Specify Entity Size to 4 to allow entities to bunch up closer than the 6 feet needed for an entity
    - When a blockage is removed, entities will automatically re-space themselves to consume 6 feet each
- **Verify effect by increasing Load/Unload times**

Statistical Analysis of Steady-State Simulations

- **Recall: Difference between terminating, steady-state simulations**
  - Which is appropriate depends on model, study
- **Now, assume steady-state is desired**
  - Be sure this is so, since running and analysis is a lot harder than for terminating simulations
- **Naturally, simulation run lengths can be long**
  - Opportunity for different internal computation order
  - Can change numerical results
  - Underscores need for statistical analysis of output

Warm Up and Run Length

- **Most models start *empty and idle***
  - *Empty*: No entities present at time 0
  - *Idle*: All resources idle at time 0
  - In a terminating simulation this is OK if realistic
  - In a steady-state simulation, though, this can *bias* the output for a while after startup
    - Bias can go either way
    - Usually downward (results are biased low) in queueing-type models that eventually get congested
    - Depending on model, parameters, and run length, the bias can be very severe

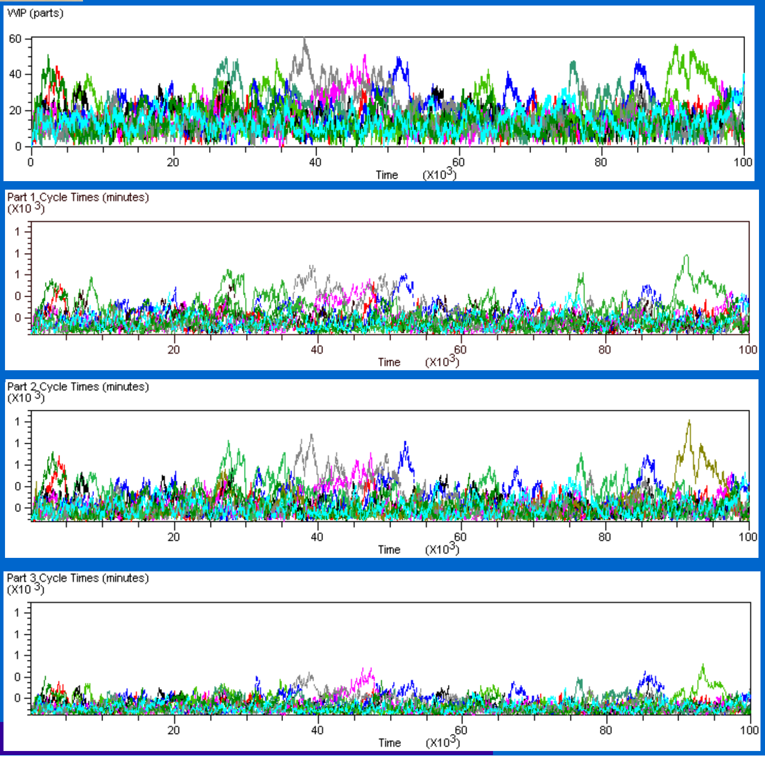
Warm Up and Run Length (cont'd.)

- **Remedies for initialization bias**
  - Better starting state, more typical of steady state
    - Throw some entities around the model
    - Can be inconvenient to do this in the model
    - How do you know how many to throw and where? (This is what you're trying to estimate in the first place.)
  - Make the run so long that bias is overwhelmed
    - Might work if initial bias is weak or dissipates quickly
  - Let model *warm up*, still starting empty and idle
    - Simulate module: Warm-Up Period (time units!)
    - "Clears" all statistics at that point for summary report, any cross-replication data saved with Statistics module's Outputs area (but not Time-Persistent or Tallies)

Warm Up and Run Length (cont'd.)

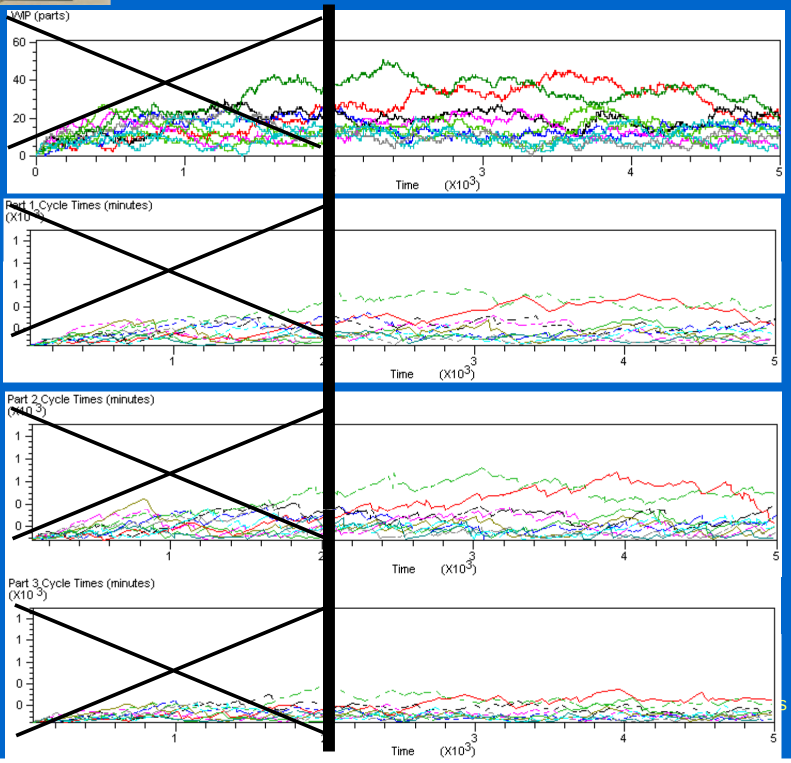
- **Warm-up and run length times?**
  - Most practical idea: preliminary runs, plots
  - Simply "eyeball" them
  - Statistics module, Time-Persistent and Tallies areas, then Plot with Output Analyzer
- **Be careful about variability — make multiple replications, superimpose plots**
- **Also, be careful to note "explosions"**
- **Model 7.4: modify Model 7.1 (Transporters)**
  - Run for 100,000 minutes, 10 replications, add WIP
  - Save within-run WIP values, Cycle Times of parts

Warm Up and Run Length (cont'd.)



- **No explosions**
- **All seem to be settling into steady state**
- **Run length seems adequate to reach steady state**
- **Hard to judge warm-up ...**

Warm Up and Run Length (cont'd.)

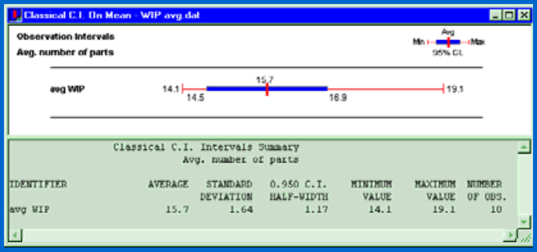


- **"Crop" plots to time 0 - 5,000**
  - Plot dialog, "Display Time from … to …"
- **Conservative warm-up: maybe 2,000**
- **If measures disagreed, use max warm-up**

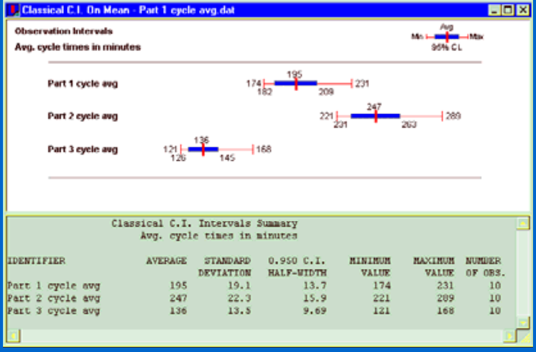
Truncated Replications

- **If you can identify appropriate warm-up and run-length times, just make replications as for terminating simulations**
  - Only difference: Specify Warm-Up Period in Simulate module
  - Proceed with confidence intervals, comparisons, all statistical analysis as in terminating case
- **Model 7.5: modify Model 7.4 (10 replications)**
  - Warm-Up Period = 2000 minutes
  - Statistics module, Outputs area entries to save summary statistics across replications

Truncated Replications (cont'd.)



```
                  Classical C.I. Intervals Summary
                        Avg. number of parts

IDENTIFIER              AVERAGE   STANDARD   0.950 C.I.   MINIMUM    MAXIMUM  NUMBER
                                  DEVIATION  HALF-WIDTH     VALUE      VALUE  OF OBS.
avg WIP                    15.7       1.64         1.17      14.1       19.1      10
```



```
                  Classical C.I. Intervals Summary
                     Avg. cycle times in minutes

IDENTIFIER              AVERAGE   STANDARD   0.950 C.I.   MINIMUM    MAXIMUM  NUMBER
                                  DEVIATION  HALF-WIDTH     VALUE      VALUE  OF OBS.
Part 1 cycle avg            195       19.1         13.7       174        231      10
Part 2 cycle avg            247       22.3         15.9       221        289      10
Part 3 cycle avg            136       13.5         9.69       121        168      10
```

- **Output Analyzer, Classical Confidence Intervals**
- **Separate invocations dues to different units**
- **Interpretation for steady-state expectations here**
- **Want smaller?**
  - More reps, same length
  - Longer reps, same number of them

Batching in a Single Run

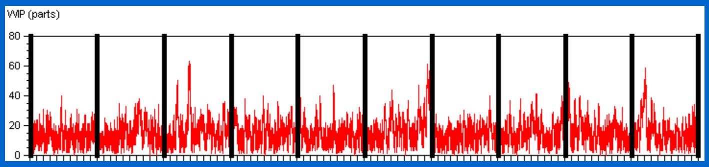
- **If model warms up very slowly, truncated replications can be costly**
  - Have to "pay" warm-up on each replication
- **Alternative: Just one *REALLY* long run**
  - Only have to "pay" warm-up once
  - Problem: Have only one "replication" and you need more than that to form a variance estimate (the basic quantity needed for statistical analysis)
  - Big no-no: Use the individual points within the run as "data" for variance estimate
    - Usually correlated (not indep.), variance estimate biased

Batching in a Single Run (cont'd.)

- **Break each output record from the run into a few large *batches***
  - Tally (discrete-time) outputs: Observation-based
  - Time-Persistent (continuous-time): Time-based
- **Take averages over batches as "basic" statistics for estimation: *Batch means***
  - Tally outputs: Simple arithmetic averages
  - Time-Persistent: Continuous-time averages
- **Treat batch means as IID**
  - Key: batch size for low correlation (details in text)
  - Still might want to truncate (once, time-based)

Batching in a Single Run (cont'd.)

- **Picture for WIP (time-persistent):**



  - For observation-based Tallies, just count points
- **To batch and analyze (details in text):**
  - Statistics module, Time-Persistent, Tally areas to save within-run records (could be big files)
  - Output Analyzer, *Analyze/Batch/Truncate* or
    - Warning if batches are too small for IID
  - Get means .flt file; Classical C.I. as before

Automatic Run-Time Confidence Intervals via Batch Means

- **Arena will automatically attempt to form 95% confidence intervals on steady-state output measures via batch means**
  - “Half Width” column in summary output
  - Ignore if you’re doing a terminating simulation
  - Uses internal rules for batch sizes (details in text)
  - Won’t report anything if your run is not long enough
    - “(Insuf)” if you don’t have the minimum amount of data Arena requires even to form a c.i.
    - “(Correl)” if you don’t have enough data to form nearly-uncorrelated batch means, required to be safe

Recommendations, Other Methods for Steady-State Analysis

- **What to do?**
  - Frankly, try to avoid steady-state simulations
    - Look at goal of the study
  - If you really do want steady-state
    - First try warm-up and truncated replications
    - Automatic run-time batch-means c.i.’s
    - Batch-means c.i.’s “by hand” if necessary
- **Other methods, goals**
  - Large literature on steady-state analysis
  - Other goals: Comparison, Selection, Ranking

Chapter 8
Detailed Modeling

By : Arya Wirabhuana

Department Of Industrial Engineering

UIN SUNAN KALIJAGA YOGYAKARTA

What We'll Do ...

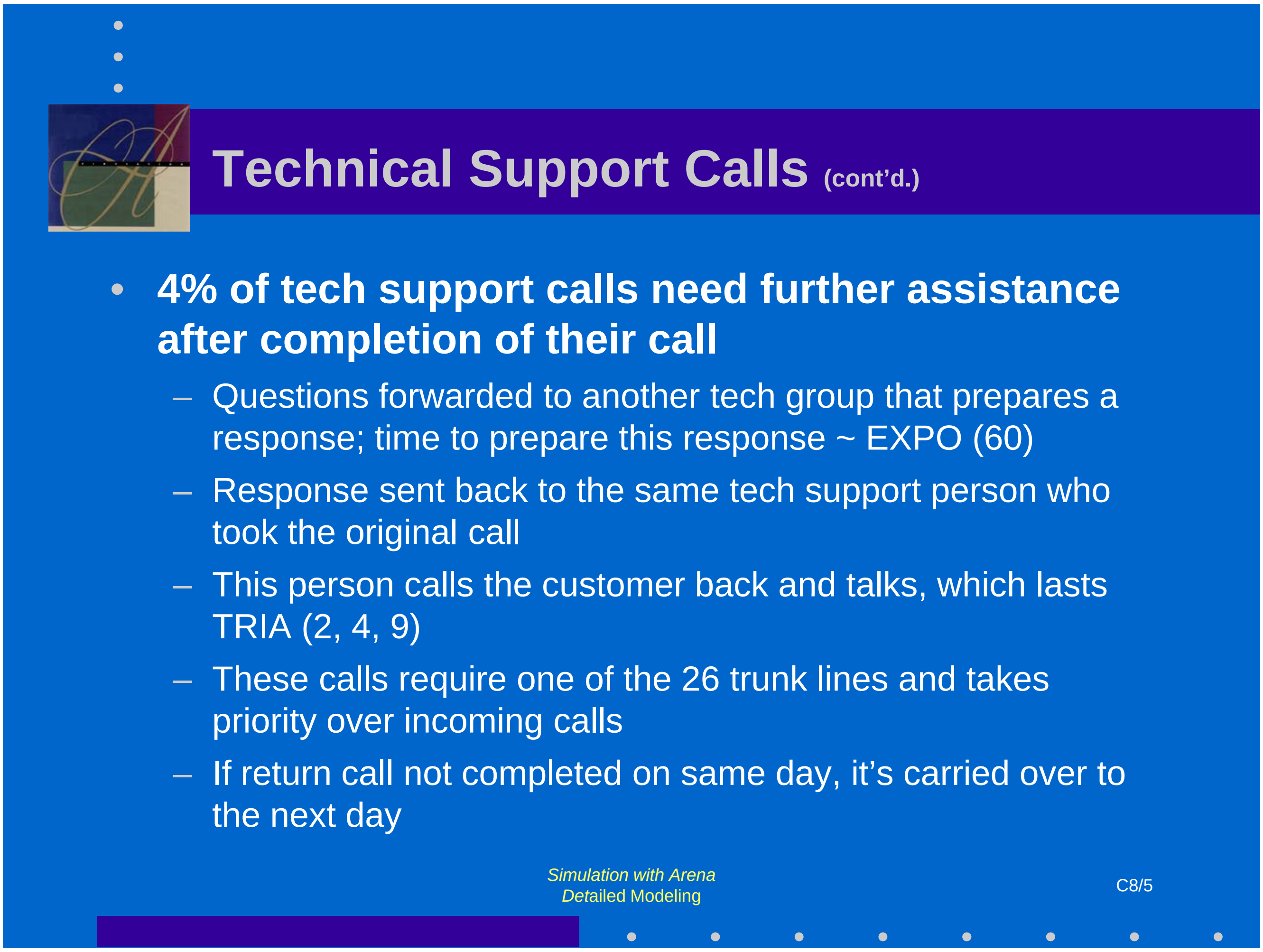
- Exploit hierarchical structure of Arena to blend in lower-level modeling for greater detail
- Example: call-center system
- Nonstationary arrival processes
- Build the model in sections, using mostly the Support panel
- Finding and fixing errors
- New kinds of more customized animations

Model 8.1: A Generic Call Center System

- **Single telephone number, 26 trunk lines**
  - If all 26 lines busy, caller gets busy signal and goes away
- **Answered call gets recording asking …**
  - Technical support? (76% of callers choose this)
  - Sales information? (16%)
  - Order-status inquiry? (8%)
- **Time for caller to choose ~ UNIF (0.1, 0.6)**
- **All times are in minutes in this model**

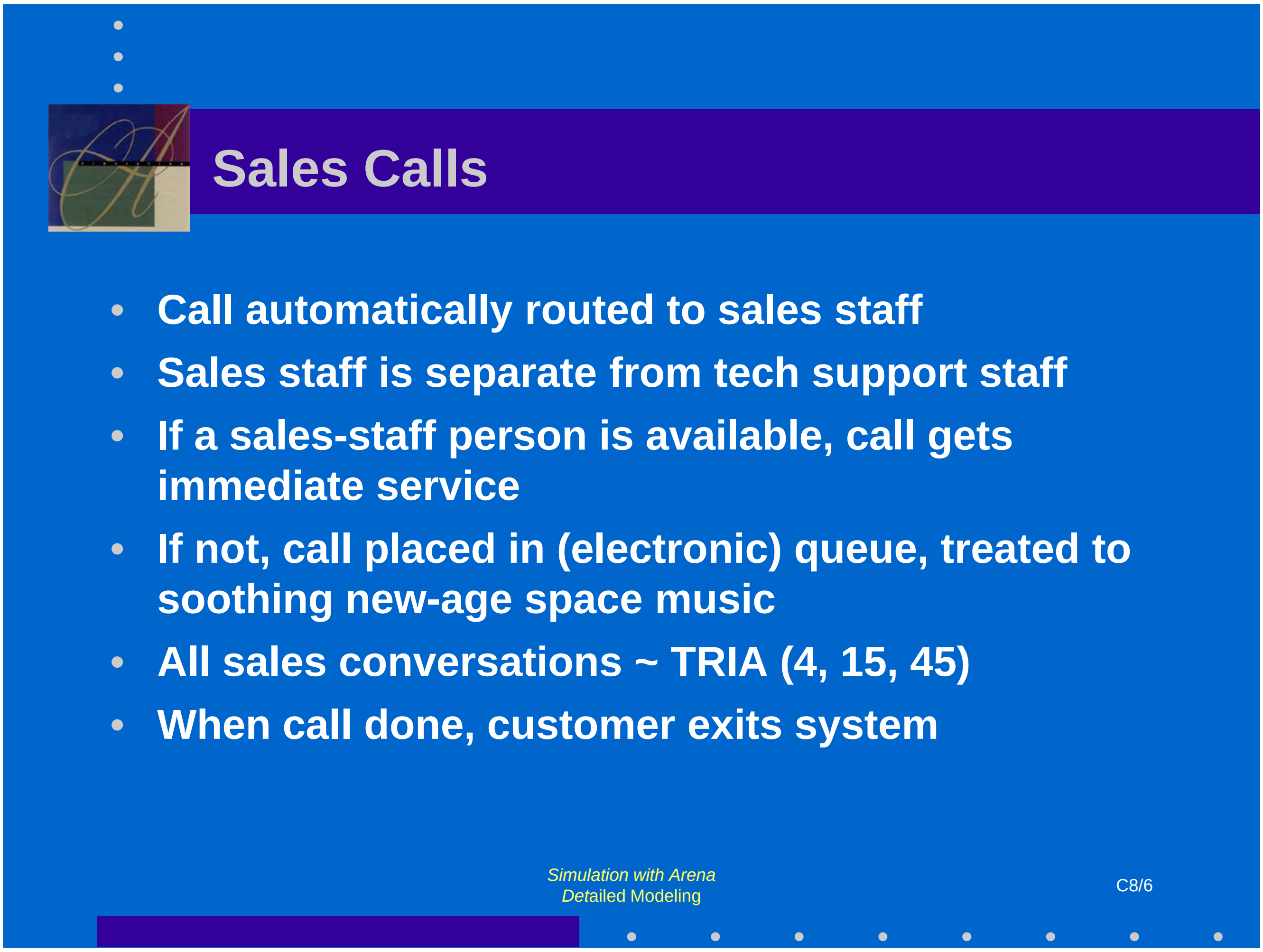
Technical Support Calls

- **Get second recording asking …**
  - Product type 1? (25% of tech support callers choose this)
  - Product type 2? (34%)
  - Product type 3? (41%)
- **If a qualified tech support person is available for chosen product, call routed for immediate service**
- **If not, call placed in (electronic) queue, subjected to annoying rock music**
- **All tech support conversations ~ TRIA (3, 6, 18)**
- **When call done, customer exits system**



Technical Support Calls (cont'd.)

- **4% of tech support calls need further assistance after completion of their call**
  - Questions forwarded to another tech group that prepares a response; time to prepare this response ~ EXPO (60)
  - Response sent back to the same tech support person who took the original call
  - This person calls the customer back and talks, which lasts TRIA (2, 4, 9)
  - These calls require one of the 26 trunk lines and takes priority over incoming calls
  - If return call not completed on same day, it's carried over to the next day



Sales Calls

- **Call automatically routed to sales staff**
- **Sales staff is separate from tech support staff**
- **If a sales-staff person is available, call gets immediate service**
- **If not, call placed in (electronic) queue, treated to soothing new-age space music**
- **All sales conversations ~ TRIA (4, 15, 45)**
- **When call done, customer exits system**

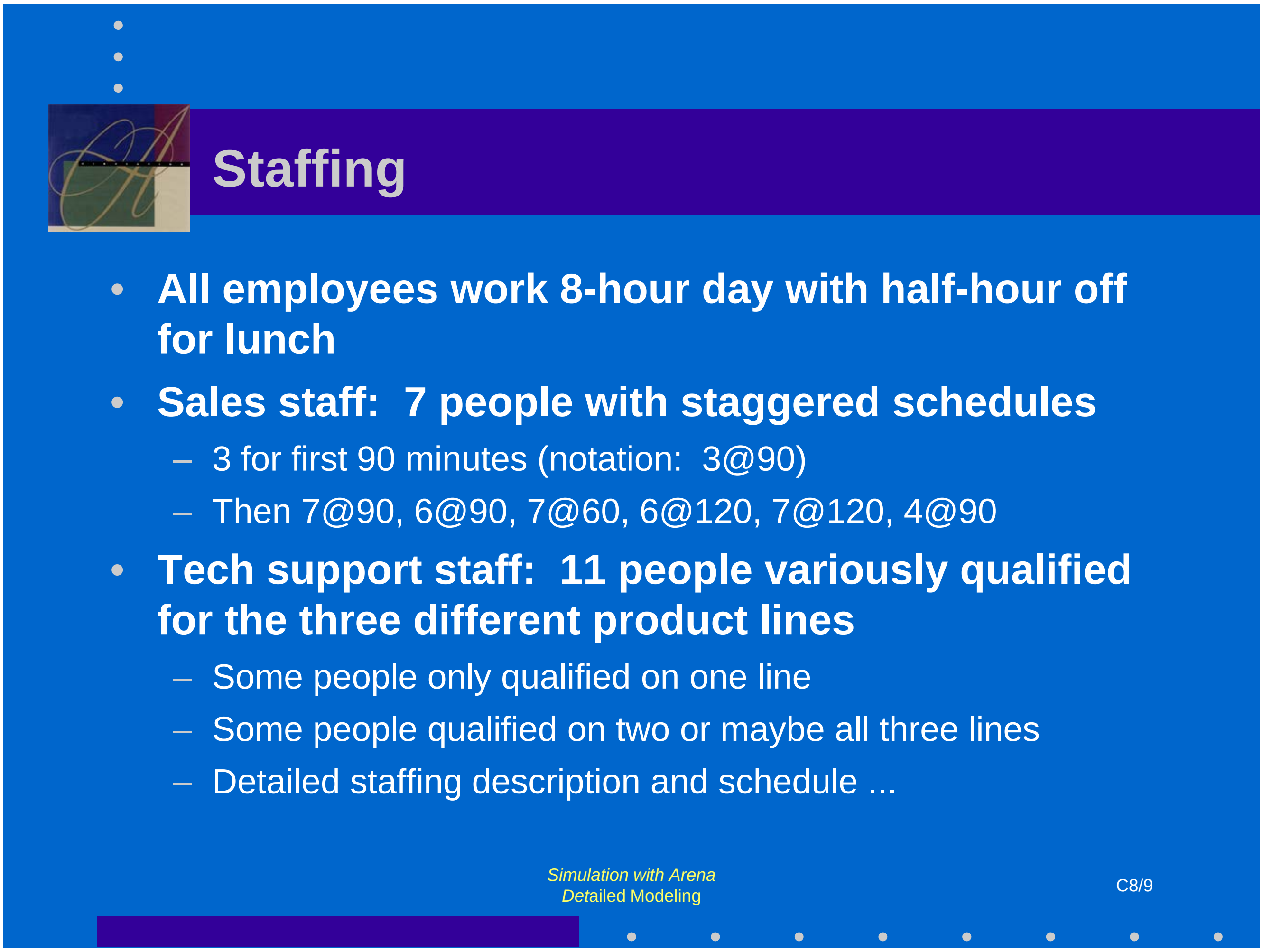
Order-Status Calls

- **Automatically handled by phone system — no people**
- **No limit on number handled at a time (but still limited by the 26 trunk lines)**
- **Time for "conversation" ~ TRIA (2, 3, 4)**
- **After call, 15% take option to talk to a real person (the rest exit the system)**
  - These calls are routed to sales staff
  - Have same priority as incoming sales calls
  - Conversation durations ~ TRIA (3, 5, 10)
  - Then exit the system

Hours and Load

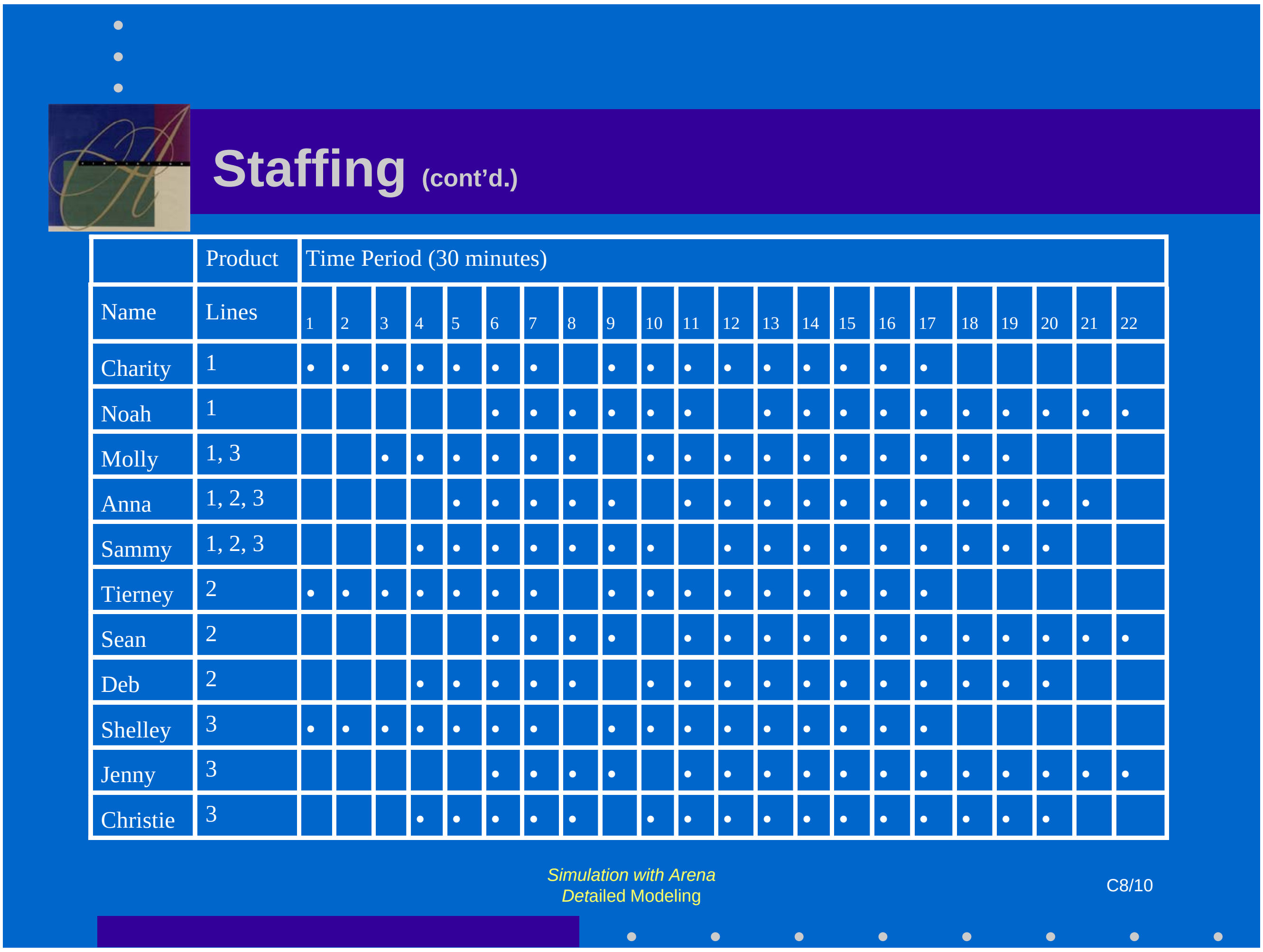
- **Operates from 8 a.m. to 6 p.m.**
- **Small proportion of the staff stays on until 7 p.m.**
  - Incoming calls shut out after 6 p.m.
  - But all calls that entered before 6 p.m. are answered
- **Call arrival rate varies substantially over the day**
  - Data on rate (calls per hour) for each half-hour period:

Time	Rate	Time	Rate	Time	Rate	Time	Rate
8:00 - 8:30	20	10:30 - 11:00	75	1:00 - 1:30	110	3:00 - 4:00	90
8:30 - 9:00	35	11:00 - 11:30	75	1:30 - 2:00	95	4:00 - 4:30	70
9:00 - 9:30	45	11:30 - 12:00	90	2:00 - 2:30	105	4:30 - 5:00	65
9:30 - 10:00	50	12:00 - 12:30	95	2:30 - 3:00	90	5:00 - 5:30	45
10:00 - 10:30	70	12:30 - 1:00	105	3:00 - 3:30	85	5:30 - 6:00	30



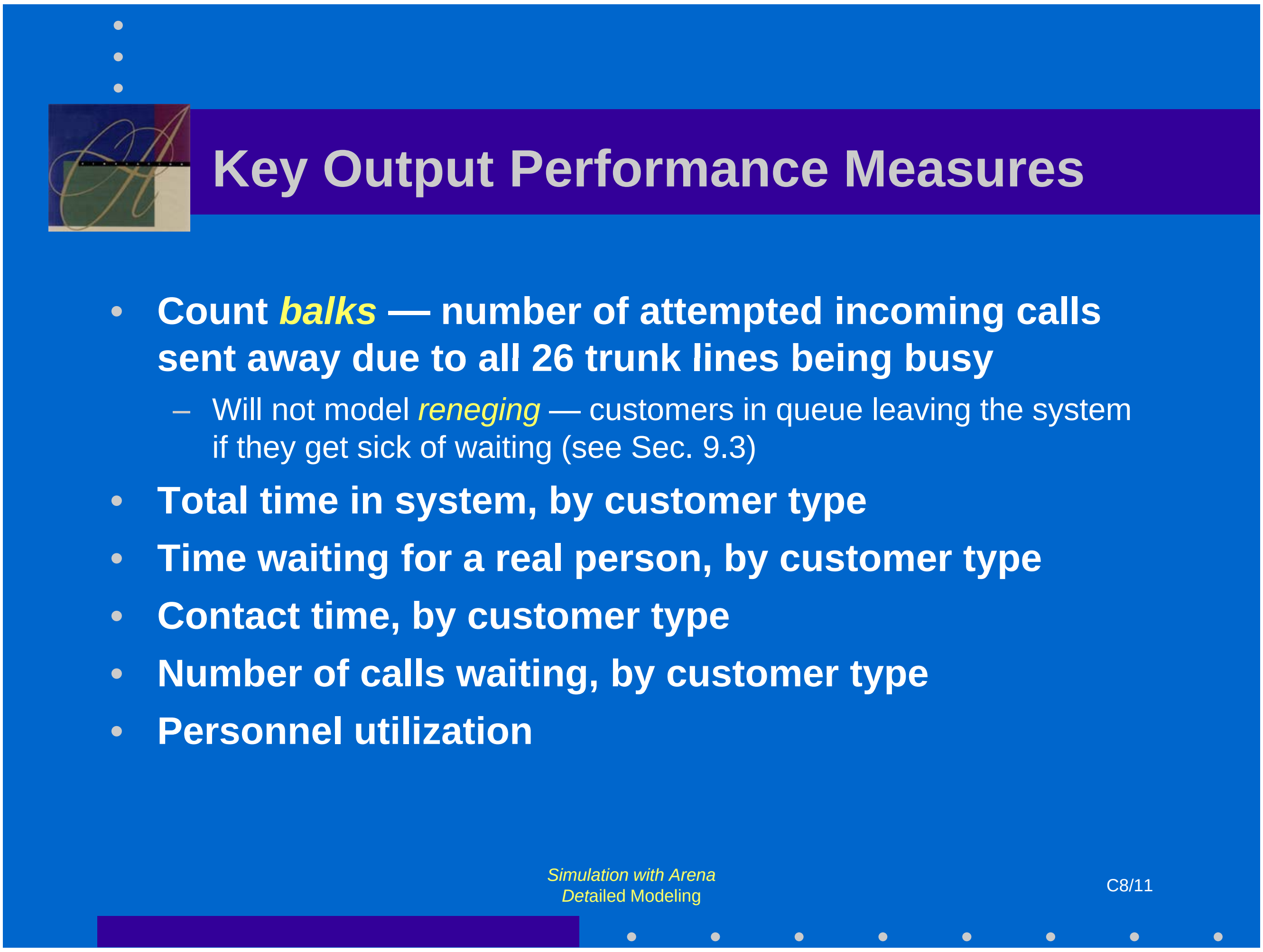
Staffing

- **All employees work 8-hour day with half-hour off for lunch**
- **Sales staff: 7 people with staggered schedules**
  - 3 for first 90 minutes (notation: 3@90)
  - Then 7@90, 6@90, 7@60, 6@120, 7@120, 4@90
- **Tech support staff: 11 people variously qualified for the three different product lines**
  - Some people only qualified on one line
  - Some people qualified on two or maybe all three lines
  - Detailed staffing description and schedule ...



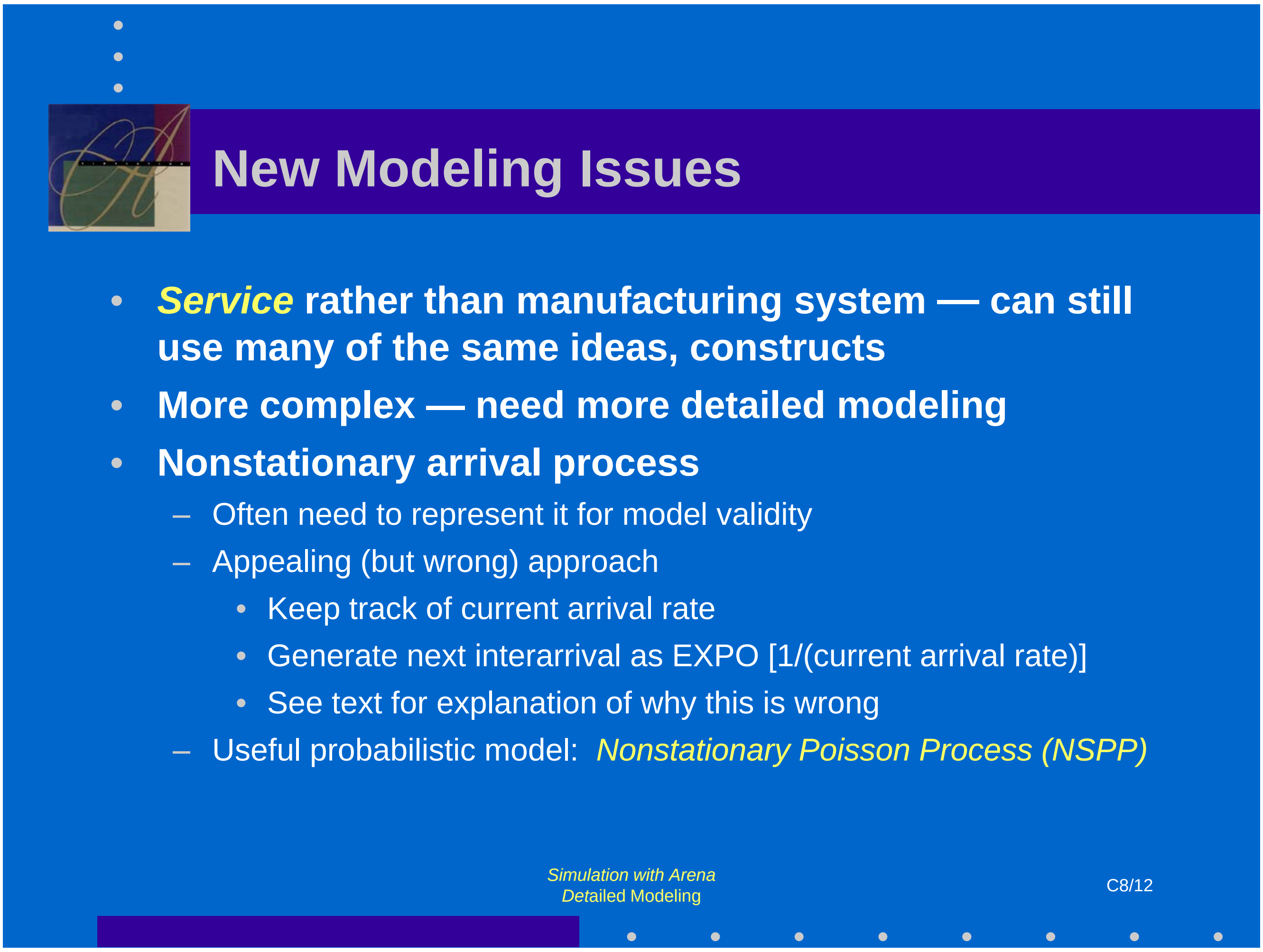
Staffing (cont'd.)

	Product	Time Period (30 minutes)																					
Name	Lines	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Charity	1	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•					
Noah	1						•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•
Molly	1, 3			•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•			
Anna	1, 2, 3					•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	
Sammy	1, 2, 3				•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•		
Tierney	2	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•					
Sean	2						•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•
Deb	2				•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•		
Shelley	3	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•					
Jenny	3						•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•
Christie	3				•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•		



Key Output Performance Measures

- **Count *balks* — number of attempted incoming calls sent away due to all 26 trunk lines being busy**
  - Will not model *reneging* — customers in queue leaving the system if they get sick of waiting (see Sec. 9.3)
- **Total time in system, by customer type**
- **Time waiting for a real person, by customer type**
- **Contact time, by customer type**
- **Number of calls waiting, by customer type**
- **Personnel utilization**



New Modeling Issues

- ***Service* rather than manufacturing system — can still use many of the same ideas, constructs**
- **More complex — need more detailed modeling**
- **Nonstationary arrival process**
  - Often need to represent it for model validity
  - Appealing (but wrong) approach
    - Keep track of current arrival rate
    - Generate next interarrival as EXPO [1/(current arrival rate)]
    - See text for explanation of why this is wrong
  - Useful probabilistic model: *Nonstationary Poisson Process (NSPP)*

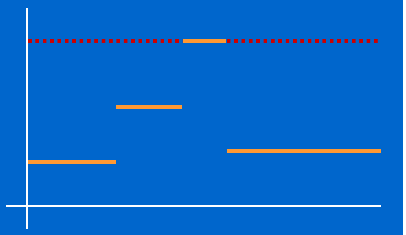
Nonstationary Poisson Process

- Calls arrive one at a time, independently, but at an expected rate that can vary over time
- Number of calls in a time interval ~ Poisson random variable with mean = area under *rate function* during the time interval
- Assume piecewise-constant rate function from data table (but form could be anything)
- One way to generate NSPP correctly: *Thinning*

NSPP Generation via Thinning

- Find max rate over the simulation time period
- Generate "candidate" arrivals with interarrival times ~ EXPO [1/(max arrival rate)]
  - Candidates are arriving too fast, except when the actual rate is at the maximum
- "Accept" a generated candidate arrival as an actual arrival with probability

$$\frac{\text{Current arrival rate}}{\text{Max arrival rate}}$$



- "Thins out" the too-fast candidates
- How to get Arena to do this?

Balking

- **Calls arriving when all 26 trunk lines are busy can’t get into system — they are lost**
- **In general, balking occurs when an entity cannot get into a queue that’s already full**
  – Limit the capacity of the queue
  – Must decide how deal with a balking entity — modeling issue
- **Here, set up a Resource for the trunk lines**
  – Give it 26 units
  – Set its queue capacity to zero
  – This is a special case of balking

Directing the Calls

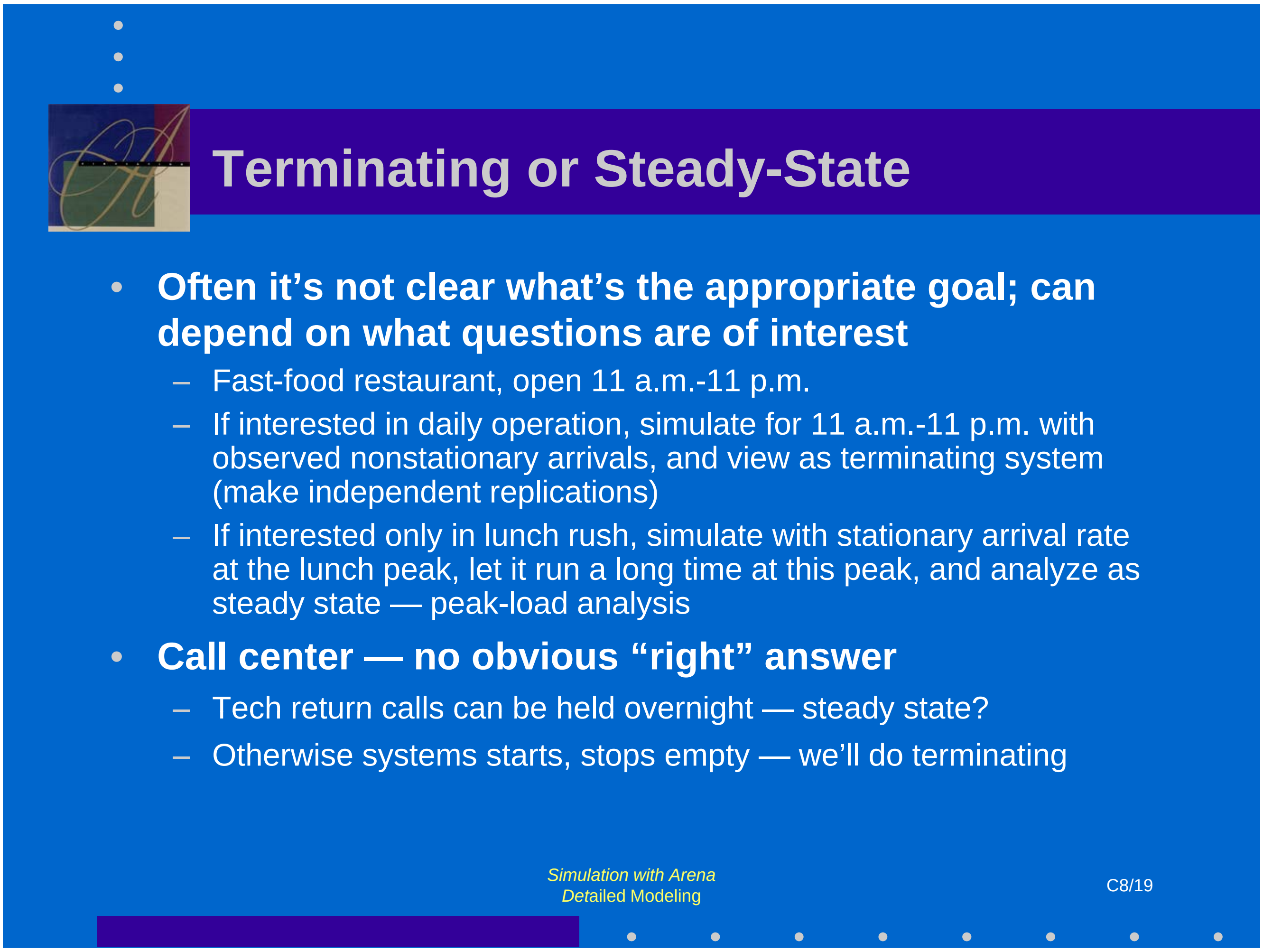
- **Have two 3-way branch points for calls**
  – Entry: Tech Support, Sales, or Order-Status Check
  – For tech support calls, Product Type 1, 2, or 3
- **Inspect module (Common panel) — only 2-way**
- **Compute final unconditional probabilities of each of the five final destinations, set up Sequences?**
  – Complicated, sensitive to future parameter changes
  – A call can control two resources — trunk line and person
  – Conclusion: this approach won’t work
- **Direct follow up: tech support calls back to same tech support person who took the original call**

Statistical and Analysis Needs

- **Type of output statistics are similar to before**
- **Analysis needs are somewhat different**
  - Maximize customer satisfaction and minimize cost — incompatible goals!
  - Customer-satisfaction measures
    - Number of busy signals (balks) — smaller is better
    - Customer wait time — smaller is better
    - Will be affected by number of trunk lines, staffing — easy to modify in model
  - Cost measures
    - Staffing schedule — try to match to varying system load
    - Usual output summaries won’t track this, so ...

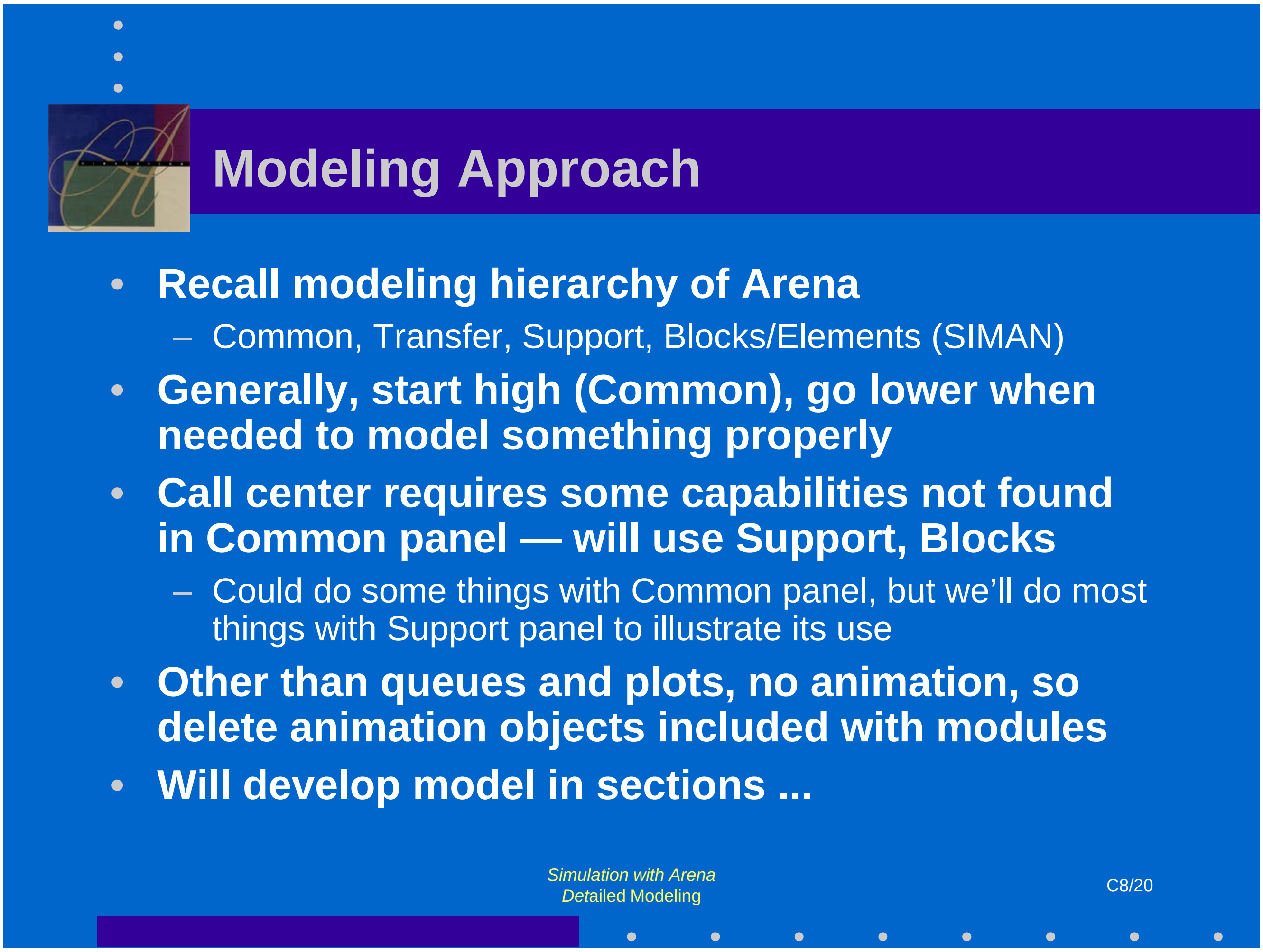
Animation Requirements

- **No entity movement to animate here**
- **Can still display queues**
- **To see how well staffing matches up with load, craft appropriate plots vs. time**
  - Number of calls balked
  - Lengths of queues
  - Number of idle staff
- **Strategy to improve performance — alter the staffing schedule, see if it produces a better matchup of the plots**



Terminating or Steady-State

- **Often it’s not clear what’s the appropriate goal; can depend on what questions are of interest**
  - Fast-food restaurant, open 11 a.m.-11 p.m.
  - If interested in daily operation, simulate for 11 a.m.-11 p.m. with observed nonstationary arrivals, and view as terminating system (make independent replications)
  - If interested only in lunch rush, simulate with stationary arrival rate at the lunch peak, let it run a long time at this peak, and analyze as steady state — peak-load analysis
- **Call center — no obvious “right” answer**
  - Tech return calls can be held overnight — steady state?
  - Otherwise systems starts, stops empty — we’ll do terminating



Modeling Approach

- **Recall modeling hierarchy of Arena**
  - Common, Transfer, Support, Blocks/Elements (SIMAN)
- **Generally, start high (Common), go lower when needed to model something properly**
- **Call center requires some capabilities not found in Common panel — will use Support, Blocks**
  - Could do some things with Common panel, but we’ll do most things with Support panel to illustrate its use
- **Other than queues and plots, no animation, so delete animation objects included with modules**
- **Will develop model in sections ...**

Defining the Data

- **Collect data modules together (Named View d)**
- **Variables — define `Arrival Rate` vector**
  - First 20 rows for half-hour rates as given
  - Last 2 rows for 6 a.m.-7 p.m., entered as 0.001 (clean up below)
- **Resource module for the 26 trunk lines (Capacity)**
- **Resource module for the sales staff (Schedule)**
- **Separate Resource modules for each tech support person (separate Schedule for each)**
- **All Schedules must cover exactly the 660 minutes (from 8 a.m.-7 p.m.) — we'll run for more than one day**

Defining the Data (cont'd.)

- **Sets module**
  - Resource sets — group tech support staff according to their capabilities on the product types
  - Tally sets — group queue-wait tallies according to product type for both original tech support calls and return calls
- **Simulate module**
  - Called for 10 replications of 660 minutes (11 hours) each
  - Two check boxes for what to reset between replications (so four possible combinations)
    - Initialize System? (We said no, due to return-call carryovers)
    - Initialize Statistics? (We said yes, to see variation across days)

Defining the Data (cont'd.)

- **Expressions module**
  - Times for tech support calls and return calls (could have done directly in modules where needed)
  - Expression for number of tech support people available (idle) for each product type — used later in animated plots

Increment Time Period

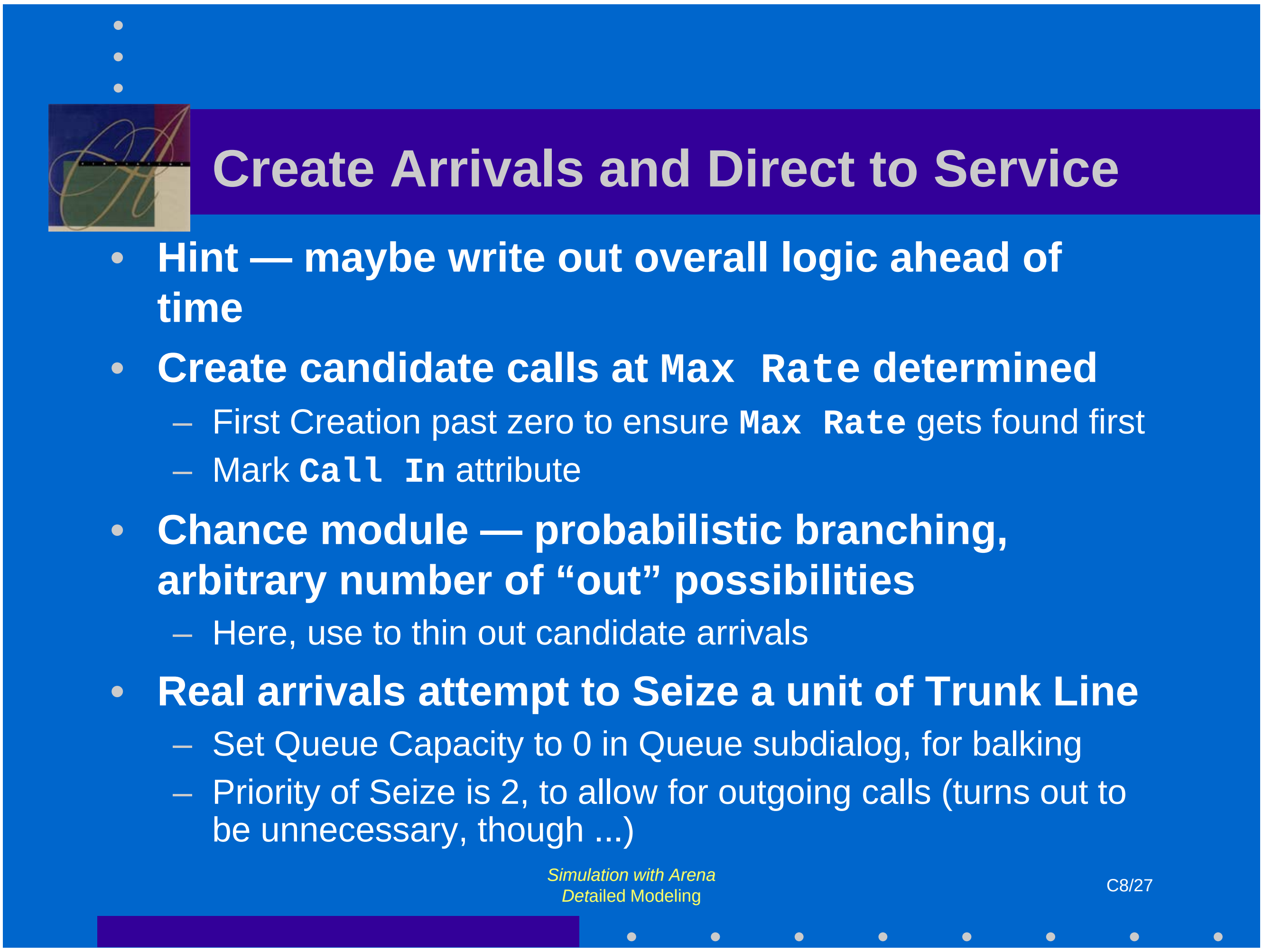
- **For NSPP thinning generation (Named View c)**
- **Generic setup — any piecewise-constant rate**
  - To indicate which half-hour period we're in
  - Global variable **`Period`** = 1, 2, …, 22 as simulation runs
- **Create module (Support) to create a "logic" entity at start of each day — Assign `Period` = 0 at start**
- **Assign module**
  - Increment **`Period`**
  - **`Per Period Balk`** = no. of balks in previous time period — used later in animated plots
  - **`Busy per Period`** = 0 (counter for balks in this period)

Increment Time Period (cont'd.)

- **Choose module — arbitrary logical branching**
  - Logical conditions on Attributes, Variables, constants
  - Notation can be like FORTRAN or C (in text)
  - Branch destinations — graphical connections or Labels
  - Each branch evaluates to true or false
    - Take First true, All trues, or All up to specified Max To Take
  - Can have > 1 copy of incoming entity sent out
  - If all branches false, incoming entity disposed, nothing out
  - This Choose module
    - If **`Period`** < 22, day not done, so Delay 30 minutes, loop back
    - If **`Period`** = 22, day done, so dispose of logic entity
- **Delay module — holds up entity for specified time**
- **Dispose module — just as it says**

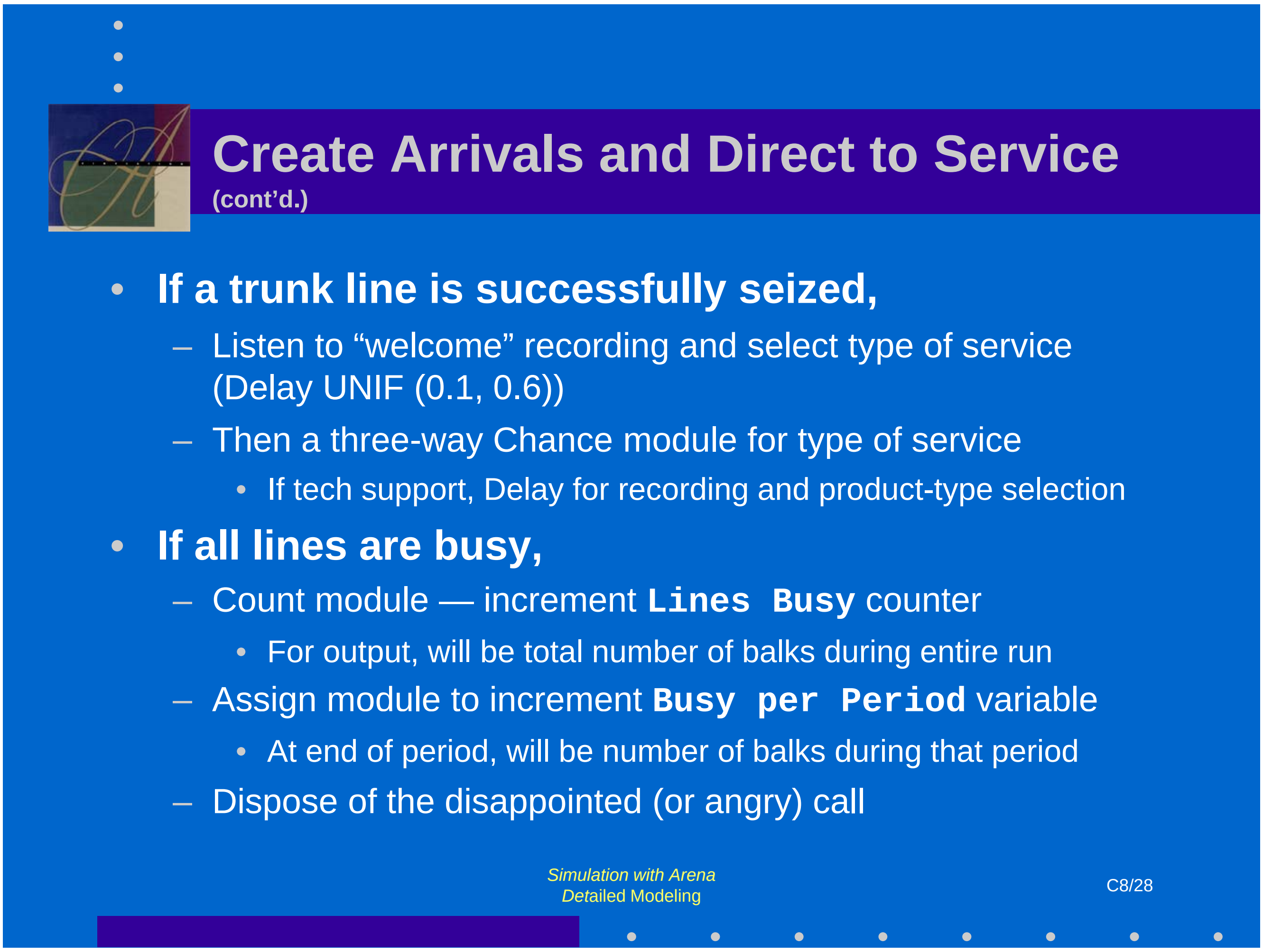
Determine Maximum Arrival Rate

- **Could do by hand and hard-code, but will put in automatic logic — generic setup (Named View m)**
- **"Code" executed just once at start of simulation**
- **Create single "finder" logic at time 0**
- **FINDJ module (Blocks panel, from SIMAN)**
  - Range of built-in Arena variable **`J`**
  - Finds first "hit" on a Condition, or min/max of Expression
  - Here, find the max in the **`Arrive Rate`** array
  - CAUTION: **`J`** is a reserved word, used for other things
- **Assign module — saves max in `Arrive Rate` array to variable `Max Rate`**
- **Dispose module — "finder" logic entity disposed**



Create Arrivals and Direct to Service

- **Hint — maybe write out overall logic ahead of time**
- **Create candidate calls at `Max Rate` determined**
  - First Creation past zero to ensure **`Max Rate`** gets found first
  - Mark **`Call In`** attribute
- **Chance module — probabilistic branching, arbitrary number of “out” possibilities**
  - Here, use to thin out candidate arrivals
- **Real arrivals attempt to Seize a unit of Trunk Line**
  - Set Queue Capacity to 0 in Queue subdialog, for balking
  - Priority of Seize is 2, to allow for outgoing calls (turns out to be unnecessary, though ...)



Create Arrivals and Direct to Service
(cont’d.)

- **If a trunk line is successfully seized,**
  - Listen to “welcome” recording and select type of service (Delay UNIF (0.1, 0.6))
  - Then a three-way Chance module for type of service
    - If tech support, Delay for recording and product-type selection
- **If all lines are busy,**
  - Count module — increment **`Lines Busy`** counter
    - For output, will be total number of balks during entire run
  - Assign module to increment **`Busy per Period`** variable
    - At end of period, will be number of balks during that period
  - Dispose of the disappointed (or angry) call

Technical Support Calls

- **Lots of modules, but many are duplicates due to three product types (Named View t)**
- **Chance module to determine product type**
- **Seize a member of resource set for product type**
  - Preferred Order rule within the set — prefer the tech support people who can only handle that type of product (reason for listing them that way in the Sets module)
  - Store index of person selected in attribute **`Ext`** for possible return for follow up call
- **Assign module — two attributes**
  - **`Product Type`** for stat collection and possible follow up
  - **`Call Start`** as Arena clock TNOW for call-duration stats

Technical Support Calls (cont'd.)

- **Delay module — from `Tech Time` Expression**
- **Release module — release resources together**
  - Tech support person — must specify resource set and the specific member in that set (**`Ext`** attribute)
  - Trunk line
- **Tally call time into product type tally within `Tech Calls` tally set**
- **Tally total time in system into an individual tally for all product types together**

Technical Support Calls (cont'd.)

- **Chance module to see if further investigation and follow up call is required — if not, call is disposed**
- **If follow up is required,**
  - Delay EXPO (6) for the investigation work itself
  - Choose module to route to **`Product Type`** by looking at the entity's attribute assigned earlier
  - Seize Specific Member of appropriate resource set (indexed by attribute **`Ext`**), with Priority 1 to override incoming calls
  - Delay module for return-call duration
  - Release tech support person
  - Tally call duration and Dispose

Sales Calls

- **Logic similar to tech support calls but simpler since there are no product types (Named View s)**
  - Seize unit of Sales Person resource
    - Use Shared Queue (see text) since this resource is also seized by Order-Status calls requesting more information
  - Assign call start time
  - Delay for call
  - Release Sales Person
  - Tally call time
  - Release Trunk Line
  - Tally time in system
  - Dispose call

Order-Status Calls

- Initially don’t require seizing of a resource, so just Assign Call Start time and delay for “conversation” (Named View s)
- Chance module to decide if customer wants to speak to a real person (sales staff)
  - If so, Seize Sales Person, Delay for call time, and Release Sales Person
- Tally call time
- Release Trunk Line
- Tally time in system
- Dispose

Reuse same modules as for Sales calls

Finding and Fixing Model Errors

- Arena picks up “simple” errors in Check phase, and leads you to them via Find and Edit buttons in Errors/Warnings windows
  - Undefined variables, attributes, stations, resources
  - Unconnected modules
  - Duplicate module names
  - Typos
- Other kinds of errors are more complex, can’t be detected without trying to run — options on Run Interaction toolbar or on Run menu
- Only mention capabilities here; see text for details

Finding and Fixing Model Errors (cont'd.)

- Run Controller — command-driven window to control, display details about model operation and underlying SIMAN code
- Trace — Follow active modules, selected variables
- Break — stop the run when entity hits a selected module, at a specific time, or when a selected entity is about to become active
- Watch — select expressions to display in a window as model runs
- Report — get summary report before run ends

Animating the Model

- No "normal" entity animation — just plots, queues, a few other "data" animations
- Plots (all vs. time)
  - Queue lengths (as in earlier models)
  - Balks per period — reason for variable `Per Period Balk`
  - Number of tech support people available for each product type — reason for the "Available" expressions defined in Expressions module
  - With multiple plots, configure first one, then copy and edit for others to get consistent look and feel; snap to grid to align
- Variable animations for Period and Day

Animating the Model (cont'd.)

- **Created digital clock "by hand" (details in text)**
  - Why not ready-made animated clocks? We didn't reset the system state between replications, so internal clock just keeps increasing
  - Resource and queue animations
  - Just for realism — doesn't add any analysis value
  - Had earlier deleted animations with modules — now restore
  - Resource button from Animate toolbar
    - Take pictures from libraries (.plb files), different states
  - Queue button from Animate toolbar
  - Add various text annotations, boxes, etc.

Animating the Model (cont'd.)

- **Storages for further animation**
  - Specified Storages in the Delay modules
  - Storages button from Animate toolbar to make them show up
  - Used for calls listening to messages, order calls (requiring no resource but only a trunk line), and number of sales calls in progress
  - Add an Entity picture via Entity button from Animate toolbar (had deleted the one that came with the Simulate module)
- **Variable for number of idle sales persons, number of available trunk lines**

TUGAS PENDAHULUAN

PRAKTIKUM PEMODELAN SISTEM

TIPE 1

1. Apa yang saudara ketahui tentang model, sistem, dan simulasi ?
2. Sebutkan dan jelaskan karakteristik model yang baik menurut Ali Basyah Siregar !
3. Sebutkan keuntungan dan kerugian dalam menggunakan simulasi !
4. Sebutkan dan jelaskan karakteristik dari model !
5. Bangkitkanlah bilangan random dengan menggunakan *Linear Congruential Method* sebanyak 10 ! (X_0 = 8, a = 12, c = 15, m = 10)

TIPE 2

1. Sebutkan dan jelaskan prinsip-prinsip pemodelan !
2. Sebutkan dan jelaskan langkah-langkah dalam melakukan simulasi !
3. Sebutkan kegunaan simulasi dalam lingkungan teknik industri !
4. Apa yang saudara ketahui tentang simulasi montecarlo ?
5. Bangkitkanlah bilangan random dengan menggunakan *Linear Congruential Method* sebanyak 10 ! (X_0 = 9, a = 15, c = 25, m = 100)

TIPE 3

1. Apa yang saudara ketahu tentang pendekatan sistem ?
2. Sebutkan dan jelaskan klasifikasi model simulasi !
3. Sebutkan perbedaan antara bilangan random dengan variabel random !
4. Sebutkan teknik-teknik pembangkitan bilangan random yang saudara ketahui !
5. Bangkitkanlah 10 bilangan random dengan menggunakan *Mid Square Method* ! (dengan X_0 = 2345)

TIPE 4

1. Apa yang saudara ketahui tentang model, sistem, dan simulasi ?
2. Sebutkan keuntungan dan kerugian dalam menggunakan simulasi !
3. Sebutkan dan jelaskan karakteristik dari bilangan random !
4. Sebutkan perbedaan antara verifikasi dengan validasi !
5. Bangkitkanlah 10 bilangan random dengan menggunakan *Mid Square Method* ! (dengan X_0 = 3456)

TUGAS PENDAHULUAN

PRAKTIKUM PEMODELAN SISTEM

TIPE 5

1. Sebutkan faktor-faktor yang mempengaruhi pemodel dalam memandang suatu model !
2. Sebutkan dan jelaskan tahap-tahap dalam pengembangan model !
3. Buatlah contoh sebuah sistem yang berada dalam lingkungan teknik industri !
4. Apa yang saudara ketahui tentang entitas, atribut, dan variabel ?
5. Bangkitkanlah bilangan random dengan menggunakan *Linear Congruential Method* sebanyak 10 ! (X_0 = 15, a = 20, c = 35, m = 100)

TIPE 6

1. Jelaskam karakteristik dan definisi dari sistem !
2. Jelaskan definisi dan klasifikasi dari model simulasi !
3. Sebutkan fungsi statistik dalam proses simulasi !
4. Sebutkan teknik-teknik pembangkitan bilangan random yang saudara ketahui !
5. Bangkitkanlah bilangan random dengan menggunakan *Linear Congruential Method* sebanyak 10 ! (X_0 = 20, a = 25, c = 40, m = 100)

TIPE 7

1. Sebutkan faktor-faktor yang mempengaruhi pemodel dalam memandang suatu model !
2. Sebutkan dan jelaskan prinsip-prinsip pemodelan !
3. Sebutkan keuntungan dan kerugian dalam menggunakan simulasi !
4. Sebutkan perbedaan antara variabel random diskret dengan variabel random kontinyu !
5. Bangkitkanlah 10 bilangan random dengan menggunakan *Mid Square Method* ! (dengan X_0 = 4567)

TIPE 8

1. Sebutkan dan jelaskan prinsip-prinsip pemodelan !
2. Buatlah contoh sebuah sistem yang berada dalam lingkungan teknik industri !
3. Sebutkan keuntungan dan kerugian dalam menggunakan simulasi !
4. Jelaskan fungsi probabilitas dalam simulasi montecarlo !
5. Bangkitkanlah 10 bilangan random dengan menggunakan *Mid Square Method* ! (dengan X_0 = 5678)

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 1

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Pada suatu pengamatan diketahu bahwa jumlah produk cacat yang keluar adalah sebagai berikut :

Jumlah Produk Cacat yang Keluar	Turus	Jumlah
0	IIIII III	8
1	IIIII II	7
2	IIII	4
3	I	1
Total	20	20

Dari diatas ujilah apakah hasil pengamatan yang dilakukan sesuai dengan distribusi binomial ! (dengan tingkat signifikansi 5%)

3. Perusahaan SIMBI Tbk yang bergerak dibidang simulasi ingin mengetahui apakah hasil simulasi dengan menambah mesin akan mampu meningkatkan jumlah produksi untuk setiap unitnya. Penelitian dilakukan dengan mengamati dan mengukur tingkat produksi sebelum dan sesudah penambahan mesin, dan dicatat data penelitian sebagai berikut :

Sebelum Penambahan Mesin	Setelah Penambahan Mesin
70	69
76	75
74	72
77	81
77	83
80	79
71	70
54	57
58	65
70	76
68	66
64	68
65	70
65	73
70	68

Dari hasil data penelitian disamping lakukanlah pengujian apakah ada peningkatan jumlah produksi dgn penambahan mesin ?

(diasumsikan data tidak mengikuti suatu distribusi apapun, terutama distribusi normal).

Lakukanlah pengujian terhadap data dengan menggunakan uji Wilcoxon, dengan nilai W_{tabel} = 25 !

UGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 2

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Dari data produksi pada sistem nyata dan data produksi hasil simulasi dibawah ini ujilah dengan uji rata-rata untuk mengetahui apakah ada perbedaan rata-rata produksi sistem nyata dengan produksi hasil simulasi ! (dengan tingkat signifikansi 5%)

Poduksi Sistem Nyata	**Produksi Hasil Simulasi**
65432	90834
79843	74132
109020	78822
29126	81419
46909	88315

3. Diketahui data bilangan random sebagai berikut :

3.11	5.81	0.51	0.31	0.31	0.71	0.51	4.01	2.91	3.81
5.79	5.89	2.99	3.49	9.09	8.39	8.19	4.89	3.39	5.69
9.28	8.68	9.08	4.08	4.28	3.28	2.78	6.08	4.58	8.18
1.27	0.67	1.87	2.97	8.77	6.67	7.67	2.37	8.37	9.87
5.26	2.76	3.76	5.26	5.76	7.36	5.56	0.76	9.96	4.06
0.15	5.35	1.45	2.65	3.85	9.15	5.35	0.75	1.75	6.25
1.44	7.34	6.34	0.94	7.64	2.84	4.74	5.94	9.84	7.74
3.33	6.43	8.43	8.73	0.33	2.13	0.63	4.03	9.63	1.53
3.92	5.62	0.22	8.32	7.62	4.42	1.62	6.02	3.62	1.92
9.31	6.51	9.61	5.71	1.41	4.31	2.11	0.11	3.61	3.51

a) Ujilah apakah data diatas mengikuti Distribusi Uniform atau tidak !

b) Ujilah apakah ada ketergantungan antar bilangan random diatas dengan menggunakan uji *Runs Test* !

Catatan : derajat ketelitian = 5%

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 3

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Pada suatu penelitian tentang sistem antrian disuatu Bank, diketahui data waktu antar kedatangan nasabah sebagai berikut :

No.	Waktu Antar Kedatangan
1	1656
2	2292
3	1268
4	2360
5	1110
6	1446
7	1520
8	1650
9	1249
10	1695

Dari data diatas ujilah apakah data waktu antar kedatangan diatas mengikuti distribusi eksponensial ? (dengan tingkat signifikansi 5%)

3. Diketahui sekumpulan bilangan random sebagai berikut :

14	57	96	40	65
76	83	02	48	56
03	87	84	60	64
76	09	63	23	73
38	26	14	07	53
57	52	37	07	27
87	29	18	59	06
42	40	90	40	86
90	34	29	60	58
03	03	05	01	58
35	36	43	31	21
19	36	44	57	16
15	86	21	92	06
77	98	28	12	47
62	17	91	52	53
40	99	73	01	55
98	83	66	14	76
81	45	32	33	81
56	33	83	39	27
38	29	07	93	05

Dari data bilangan random disamping, ujilah apakah data tersebut mengikuti Distribusi Normal ! (dengan tingkat signifikansi 5%)

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 4

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Dari data jumlah kedatangan dengan periode setiap 5 menit dibawah ini, ujilah data pengamatan dibawah ini dengan uji poisson. Kemudian amati apakah data pengamatan yang dibangkitkan sesuai dengan bentuk Distribusi Poisson ! (dengan tingkat signifikansi 5%)

Kedatangan per Periode	Frekuensi	Kedatangan per Periode	Frekuensi
0	12	6	7
1	10	7	5
2	19	8	5
3	17	9	3
4	10	10	3
5	8	11	1

3. Dalam suatu penelitian, diketahui data waktu proses mesin sebagai berikut :

12.60	13.60	14.54	14.94	15.09	15.38	15.64	15.77	16.30	16.83
12.70	13.74	14.59	14.95	15.18	15.38	15.68	15.84	16.42	17.44
12.80	13.80	14.85	15.08	15.33	15.58	15.68	16.24	16.47	17.85

Untuk penentuan distribusi probabilitas yang akan dibandingkan dengan distrbusi probabilitas empiris, maka akan dilakukan uji kesesuaian dengan distribusi toritis yaitu distribusi normal.. Data teoritis dibangkitkan dengan menggunakan salah satu software, dan diketahui datanya sebagai berikut :

12.02	13.70	14.39	14.73	14.92	15.71	16.07	16.22	16.44	17.74
13.20	13.71	14.40	14.73	15.03	15.86	16.13	16.37	16.55	17.50
13.46	13.97	14.49	14.87	15.06	15.91	16.13	16.42	16.59	17.68

Lakukanlah Uji Kesesuaian/Goodness of Fit dengan menggunakan metode Kolmogorov-Smirnov ! (dengan tingkat signifikansi 5%)

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 5

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Dari data bilangan random dibawah ini, ujilah apakah data bilangan random tersebut mengikuti Distribusi Normal ! (dengan tingkat signifikansi 5%)

93	45	57	65	96	02	56	83	68	39
23	87	64	84	63	73	09	12	14	33
40	09	29	05	07	83	81	33	34	48
60	12	45	27	32	66	76	83	20	23
01	78	26	53	14	52	63	52	27	37
12	65	29	06	18	90	86	40	92	92
07	07	64	82	17	99	53	55	91	73
40	59	13	18	86	98	06	47	21	28
01	69	36	21	43	44	16	36	72	60
31	08	03	58	05	29	58	34	12	57

3. Kuis matakuliah Statistik diberikan kepada sebanyak 20 mahasiswa kelas A, dan kuis yang sama juga diberikan kepada 15 mahasiswa kelas B. Dari hasil kuis yang diperoleh di dua kelas tersebut menunjukkan nilai kuis tiap mahasiswanya sebagai berikut :

Mahasiswa	Nilai Kelas A	Nilai Kelas B
1	70	72
2	63	67
3	78	56
4	71	69
5	82	71
6	93	59
7	96	55
8	61	88
9	72	79
10	63	49
11	56	76
12	82	53
13	66	66
14	76	73
15	67	80
16	61	
17	74	
18	86	
19	64	
20	93	

Dari data disamping, lakukanlah pengujian untuk mengetahui apakah ada perbedaan tingkat kepandaian pada kedua kelas tersebut !

Lakukanlah pengujian dengan Uji Mann-Whitney Test. (dengan α = 5%)

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 6

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Dari data tersebut dibawah ini :
   a. Ujilah apakah data tersebut mengikuti Distribusi Uniform atau tidak !
   b. Ujilah apakah ada ketergantungan antar bilangan random tersebut dengan menggunakan uji tanda !0.64

0.32	0.98	0.78	0.19	0.32	0.05	0.58	0.29	0.09	0.03
0.03	0.36	0.01	0.08	0.36	0.81	0.58	0.33	0.36	0.42
0.21	0.84	0.03	0.15	0.87	0.23	0.15	0.60	0.33	0.40
0.28	0.63	0.76	0.77	0.09	0.27	0.86	0.45	0.14	0.59
0.73	0.37	0.57	0.40	0.52	0.76	0.73	0.83	0.52	0.07
0.66	0.18	0.87	0.98	0.29	0.55	0.27	0.99	0.12	0.23
0.32	0.90	0.42	0.81	0.40	0.47	0.06	0.98	0.92	0.60
0.50	0.45	0.04	0.12	0.10	0.06	0.64	0.86	0.02	0.87
0.83	0.29	0.90	0.56	0.34	0.87	0.95	0.23	0.57	0.48
0.07	0.05	0.03	0.38	0.03	0.30	0.25	0.24	0.31	0.40

(dengan α = 5%)

3. Untuk membandingkan tingkat keefektifan dari 3 macam metode untuk menurunkan dan menjaga keseimbangan berat badan, maka sebanyak 22 orang mahasiswa FTI – UII dibagi kedalam tiga (3) kelompok (metode I = 6, metode II = 7, dan metode III = 9), dimana masing-masing kelompok mengikuti program diet selama 4 minggu sesuai dengan metode yang ada diatas. Setelah program diet berakhir, maka diperoleh data banyaknya berat badan yang hilang dari mahasiswa-mahasisw tersebut sebagai berikut :

Jumlah Berat Badan yang Hilang (Kg)		
Metode I	Metode II	Metode III
5,3	6,3	2,4
4,2	8,4	3,1
3,7	9,3	3,7
7,2	6,5	4,1
6,0	7,7	2,5
4,8	8,2	1,7
	9,5	5,3
		4,5
		1,3

Gunakanlah metode Kruskal-Wallis, untuk menguji apakah ada perbedaan tingkat keefektifan dari ketiga metode diet diatas ! (dengan α = 5%)

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 7

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Dari data tersebut dibawah ini :
   a. Ujilah apakah data tersebut mengikuti Distribusi Uniform atau tidak !
   b. Ujilah apakah ada ketergantungan antar bilangan random tersebut dengan menggunakan uji tanda !0.64

0.43	0.44	0.21	0.28	0.73	0.66	0.07	0.42	0.50	0.29
0.96	0.02	0.84	0.63	0.37	0.18	0.05	0.81	0.04	0.33
0.21	0.16	0.06	0.47	0.55	0.76	0.05	0.40	0.12	0.57
0.65	0.56	0.64	0.73	0.27	0.06	0.58	0.45	0.45	0.48
0.01	0.60	0.40	0.59	0.07	0.23	0.40	0.60	0.10	0.58
0.93	0.39	0.33	0.14	0.52	0.12	0.31	0.86	0.23	0.81
0.36	0.36	0.86	0.98	0.99	0.83	0.29	0.92	0.15	0.34
0.57	0.83	0.87	0.09	0.52	0.29	0.03	0.27	0.02	0.56
0.35	0.19	0.15	0.77	0.40	0.98	0.38	0.90	0.60	0.90
0.14	0.76	0.03	0.76	0.57	0.87	0.03	0.32	0.87	0.83

(dengan α = 5%)

3. Dari data bilangan random dibawah ini, ujilah apakah data bilangan random tersebut mengikuti Distribusi Normal ! (dengan tingkat signifikansi 5%)

39	93	96	02	68	56	57	83	65	45
33	23	63	73	14	09	64	12	84	87
48	40	07	83	34	81	29	33	05	09
23	60	32	66	20	76	45	83	27	12
37	01	14	52	27	63	26	52	53	78
92	12	18	90	92	86	29	40	06	65
73	07	17	99	91	53	64	55	82	07
28	40	86	98	21	06	13	47	18	59
60	01	43	44	72	16	36	36	21	69
57	31	05	29	12	58	03	34	58	08

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 1

TIPE 8

1. Sebutkan dan jelaskan macam-macam distribusi data yang saudara ketahui !
2. Diketahui data bilangan random sebagai berikut :

0.60	0.84	0.70	0.30	0.26	0.38	0.05	0.19	0.73	0.44
0.95	0.27	0.41	0.81	0.96	0.31	0.09	0.06	0.23	0.77
0.48	0.86	0.14	0.86	0.89	0.37	0.49	0.60	0.04	0.83
0.73	0.47	0.13	0.55	0.11	0.75	0.36	0.25	0.23	0.72
0.30	0.48	0.36	0.01	0.54	0.34	0.96	0.06	0.61	0.85
0.42	0.83	0.37	0.21	0.90	0.89	0.91	0.79	0.57	0.99

Ujilah data bilangan random diatas dengan menggunakan metode *Length of Runs Up and Down* atau *Length of runs Above and Bellow the Mean*, untuk mengetahui bahwa bilangan random yang dibangkitkan mempunyai sifat/karakteristik yang independent ! (dengan α = 5%)

3. Seorang Ketua Jurusan ingin mengetahui apakah ada perbedaan tinggi rendahnya tingkat IQ asisten antara Laboratorium X dengan Laboratorium Y. Untuk itu diambillah secara random sebanyak 14 asisten Laboratorium X dan 12 asisten Laboratorium Y. Dari hasil pengetesan diperoleh hasil sebagai berikut :

Asisten	Hasil Test Asisten Laboratorium X	Hasil Test Asisten Laboratorium Y
1	130	111
2	128	116
3	119	124
4	125	109
5	120	105
6	132	127
7	118	130
8	110	125
9	126	103
10	123	122
11	115	101
12	129	110
13	125	
14	133	

Dari data disamping, lakukanlah pengujian untuk mengetahui apakah ada perbedaan tingkat kepandaian antar asisten pada kedua Laboratorium tersebut !

Lakukanlah pengujian dengan Uji Mann-Whitney Test. (dengan α = 5%)

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 2

TIPE 1

Seorang distributor surat kabar dan majalah sedang mempertimbangkan berapa surat kabar yang sebaiknya dipesan per hari dengan pertimbangan pertama adalah pertimbangan dari tipe beritanya. Ada tiga tipe berita yang mungkin akan muncul pada hari tersebut yaitu bagus, sedang, buruk. Pertimbangan kedua adalah permintaannya. Fluktuasi permintaan antara 50 sampai 100 unit surat kabar per hari. Pihak supplier hanya melayani pembelian surat kabar dengan kelipatan 10 unit dengan harga per unit Rp 1000,-. Distributor menjual surat kabar tersebut juga dengan kelipatan 10 dengan harga Rp 1500,- per unit. Surat kabar yang tidak laku pada pagi hari akan dijual pada sore atau malam harinya dengan harga Rp 500,- per unit. Dari data masa lalu diketahui bahwa materi berita dengan permintaan masing-masing adalah bersifat independen dan diperoleh probabilitas dari materi berita dan permintaan dapat dilihat pada tabel berikut :

Untuk materi berita mempunyai probabilitas sebagai berikut :

Materi Berita	**Probabilitas**
Bagus	0,40
Sedang	0,35
Buruk	0,25

Untuk probabilitas permintaan mempunyai probabilitas sebagai berikut :

Permintaan	Probabilitas Permintaan		
	Bagus	Sedang	Buruk
50	0,20	0,15	0,12
60	0,15	0,25	0,18
70	0,25	0,10	0,20
80	0,10	0,20	0,25
90	0,17	0,14	0,15
100	0,13	0,16	0,10

Dari kondisi diatas pihak distributor meminta bantuan saudara untuk memberikan gambaran tentang berapa jumlah surat kabar yang harus dipesan sehingga akan memberikan keuntungan yang maksimal.

Simulasi dilakukan untuk mengetahui fluktuasi permintaan selama 1 hari dengan 30 replikasi.

1. Buatlah flowchart penyelesaian masalah diatas !
2. Buatlah format tabel untuk menyelesaikan kasus diatas !

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 2

TIPE 2

Sebuah sistem antrian yang terdiri dari satu buah stasiun pelayanan, diketahui mempunyai waktu antar kedatangan konsumen antara 1 – 5 menit. Sedangkan waktu pelayanan petugas antara 1 – 6 menit, dengan probabilitas yang ditunjukkan pada tabel dibawah ini :

Waktu Antar Kedatangan	Probabilitas
1	0,20
2	0,15
3	0,15
4	0,20
5	0,30

Waktu Pelayanan	Probabilitas
1	0,10
2	0,20
3	0,25
4	0,15
5	0,10
6	0,20

1. Lakukanlah perhitungan dengan menggunakan simulasi monte carlo untuk 10 pelanggan, sehingga kita bisa mengetahui :
   a. Total waktu pelayanan
   b. Total waktu antrian
   c. Total waktu menganggur petugas
   d. Total waktu pelanggan berada dalam sistem
2. Lakukanlah analisa terhadap sistem tersebut !

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 2

TIPE 3

Suatu perusahaan yang bergerak dibidang manufaktur, pada bagian produksinya mempunyai berbagai macam jenis mesin yang digunakan dalam proses produksinya. Salah satu diantaranya adalah mesin bubut, dimana perusahaan tersebut mempunyai dua buah mesin bubut. Sehingga perusahaan menggunakan sistem *multi server single channel*, dengan demikian mesin bubut 2 akan bekerja jika mesin bubut 1 tidak kosong (bekerja). Adapun data pengamatan mengenai waktu antar kedatangan produk dan waktu pelayanan masing-masing mesin adalah sebagai berikut

Waktu Antar Kedatangan	Probabilitas
1	0,20
2	0,15
3	0,25
4	0,10
5	0,30

Waktu Pelayanan Mesin Bubut 1	Probabilitas	Waktu Pelayanan Mesin Bubut 2	Probabilitas
3	0,30	4	0,30
4	0,35	5	0,35
5	0,35	6	0,25
-	-	7	0,10

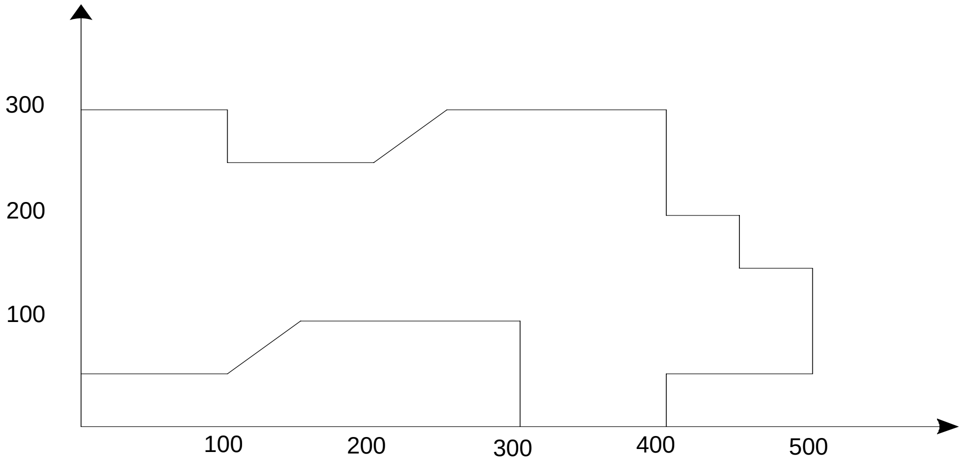
1. Lakukanlah perhitungan dengan menggunakan simulasi monte carlo untuk 10 produk, sehingga kita bisa mengetahui :
   a. Total waktu pelayanan masing-masing mesin.
   b. Total waktu antrian dari produk.
2. Lakukanlah analisa terhadap sistem tersebut !

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 2

TIPE 4

1. Sebuah pesawat pengebom akan menjatuhkan bom pada jarak horisontal 500 m dan vertikal 300 m, dengan gambar lokasi sasaran seperti dibawah ini :



   Apabila pesawat tersebut menjatuhkan 15 bom secara acak, berapakah jumlah bom yang akan mengenai sasaran ? (lakukanlah perhitungan diatas dengan menggunakan simulasi monte carlo dengan koordinat sasaran bom diasumsikan mengikuti distribusi normal).

2. Tentukanlah dengan menggunakan teknik simulasi monte carlo nilai dari Z, jika diketahui data sebagai berikut :

$$Z = \sqrt{\frac{(2\,A + B}{4\,C}}$$

   A = Normal (25, 5)

   B = Uniform (11, 15)

   C =

C_i	d_i
10	0,30
15	0,20
20	0,25
25	0,25

TUGAS PENDAHULUAN

PRAKTIKUM EXCEL 2

TIPE 5

Seorang pedagang buah ingin mengetahui berapa jumlah buah yang harus dipesan pada setiap harinya sehingga mampu memberikan keuntungan yang besar. Ada empat macam jenis buah yang akan dijual setiap harinya yaitu jeruk, apel, mangga, dan anggur. Adapun jumlah permintaan setiap harinya antara 1 sampai 10 kg. Pedagang hanya menjual buah setiap kelipatan 1 kg. Adapun daftar harga dari petani dan pedagang untuk masing-masing buah adalah sebagai berikut :

Jenis Buah	Harga	
	Petani	Pedagang
Jeruk	Rp 5000,-	Rp 5500,-
Apel	Rp 7500,-	Rp 7750,-
Mangga	Rp 4000,-	Rp 5000,-
Anggur	Rp 6000,-	Rp 6500,-

Dari data masa lalu diketahui bahwa jenis buah dengan permintaan masing-masing adalah bersifat independen dan diperoleh probabilitas dari jenis buah dan permintaan dapat dilihat pada tabel berikut :

Untuk jenis buah mempunyai probabilitas sebagai berikut :

Materi Berita	**Probabilitas**
Jeruk	0,30
Apel	0,25
Mangga	0,25
Anggur	0,20

Untuk probabilitas permintaan mempunyai probabilitas sebagai berikut :

Permintaan	Probabilitas Permintaan			
	Jeruk	Apel	Mangga	Anggur
1	0,05	0,10	0,05	0,10
2	0,10	0,15	0,05	0,15
3	0,15	0,05	0,10	0,05
4	0,10	0,10	0,15	0,10
5	0,20	0,10	0,05	0,10
6	0,05	0,20	0,10	0,20
7	0,05	0,05	0,10	0,05
8	0,10	0,05	0,05	0,05
9	0,10	0,05	0,10	0,05
10	0,10	0,10	0,20	0,10

Dari kondisi diatas pihak pedagang meminta saudara untuk memberikan gambaran tentang berapa jumlah masing-masing buah yang harus dipesan sehingga akan memberikan jumlah keuntungan yang maksimal.

Simulasi dilakukan dengan tujuan untuk mengetahui fluktuasi permintaan selama 1 hari dengan 20 replikasi.

1. Buatlah flowchart untuk penyelesaian masalah diatas !
2. Buatlah format tabel (kolom) untuk penyelesaian kasus diatas !

TUGAS PENDAHULUAN

PRAKTIKUM ARENA 1

TIPE 1

1. Apa yang saudara ketahui tentang Arena ?
2. Sebutkan dua (2) jenis bahasa pemrograman yang dipakai dalam simulasi !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

   Data Waktu Antar Kedatangan = Distribusi Eksponensial (5)

   Data Waktu Proses = Distribusi Poisson (4)

   Kemudian lakukanlah analisis dari hasil proses simulasi !

TIPE 2

1. Sebutkan dan jelaskan jenis-jenis software Simulasi yang saudara ketahui !
2. Dalam bahasa simulasi dikenal dengan istilah event orientation dan process orientation. Jelaskan perbedaan dari keduanya !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

   Data Waktu Antar Kedatangan = Distribusi Uniform (5, 7)

   Data Waktu Proses = Distribusi Normal (5, 2)

   Kemudian lakukanlah analisis dari hasil proses simulasi !

TUGAS PENDAHULUAN

PRAKTIKUM ARENA 1

TIPE 3

1. Sebutkan kelebihan software Arena dibandingkan dengan software yang lain !
2. Sebutkan aplikasi yang bisa dilakukan dengan menggunakan software Arena !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

   Data Waktu Antar Kedatangan = Distribusi Triangular (3, 5, 7)

   Data Waktu Proses = Distribusi Poisson (6)

   Kemudian lakukanlah analisis dari hasil proses simulasi !

TIPE 4

1. Sebutkan perbedaan bahasa simulasi General Purpose Simulation Language (GPSL) dengan Special Purpose Simulation Language (SPSL) !
2. Sebutkan kelemahan dari program simulasi !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

   Data Waktu Antar Kedatangan = Distribusi Eksopnensial (7)

   Data Waktu Proses = Distribusi Triangular (4, 6, 8)

   Kemudian lakukanlah analisis dari hasil proses simulasi !

TUGAS PENDAHULUAN

PRAKTIKUM ARENA 1

TIPE 5

1. Jelaskan langkah-langkah masuk ke dalam Arena !
2. Jelaskan langkah-langkah membuat file dalam Arena !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

   Data Waktu Antar Kedatangan = Distribusi Eksopnensial (10)

   Data Waktu Proses = Distribusi Normal (10,3)

   Kemudian lakukanlah analisis dari hasil proses simulasi !

TIPE 6

1. Apa yang saudara ketahui tentang Arena ?
2. Sebutkan kelebihan software Arena dibandingkan dengan software yang lain !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

   Data Waktu Antar Kedatangan = Distribusi Eksopnensial (7)

   Data Waktu Proses = Distribusi Normal (6,3)

   Kemudian lakukanlah analisis dari hasil proses simulasi !

TUGAS PENDAHULUAN

PRAKTIKUM ARENA 1

TIPE 7

1. Sebutkan perbedaan bahasa simulasi General Purpose Simulation Language (GPSL) dengan Special Purpose Simulation Language (SPSL) !
2. Sebutkan aplikasi yang bisa dilakukan dengan menggunakan software Arena !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

Data Waktu Antar Kedatangan	= Distribusi Triangular (5, 7, 9)
Data Waktu Proses	= Distribusi Poisson (8)

   Kemudian lakukanlah analisis dari hasil proses simulasi !

TIPE 8

1. Dalam bahasa simulasi dikenal dengan istilah event orientation dan process orientation. Jelaskan perbedaan dari keduanya !
2. Jelaskan langkah-langkah masuk ke dalam Arena !
3. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Arrive
   - Server
   - Depart
4. Buatlah contoh program Arena dengan menggunakan modul-modul diatas !

   Dengan data-data yang digunakan adalah sebagai berikut :

Data Waktu Antar Kedatangan	= Distribusi Uniform (8, 12)
Data Waktu Proses	= Distribusi Normal (9, 2)

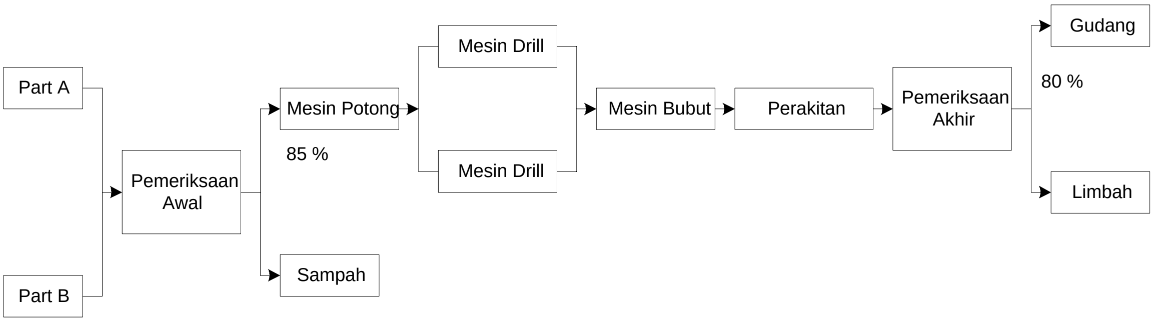
   Kemudian lakukanlah analisis dari hasil proses simulasi !

TUGAS PENDAHULUAN

PRAKTIKUM ARENA 2

TIPE 1

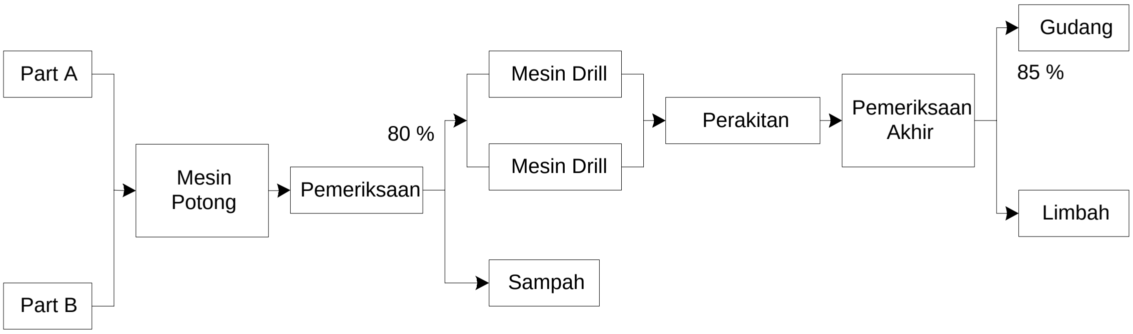
1. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Sequence
   - AdvServer
   - Queue
   - Set
   - Resource
   - Expression
2. Buatlah model dibawah ini dengan menggunakan software Arena !



Dengan data waktu antar kedatangan, waktu proses (khusus untuk pemeriksaan akhir masing-masing part beda), dan waktu perjalanan ***bebas*** (simulasi 8 jam)

TIPE 2

1. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Sequence
   - AdvServer
   - Queue
   - Set
   - Resource
   - Expression
2. Buatlah model dibawah ini dengan menggunakan software Arena !



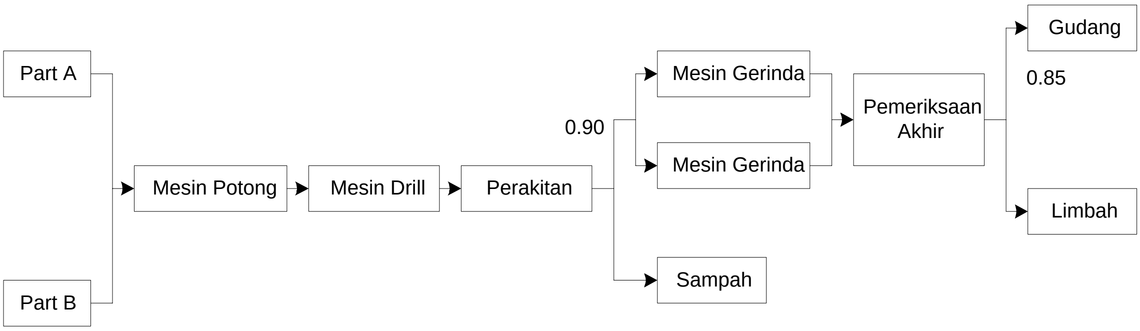
Dengan data waktu antar kedatangan, waktu proses (khusus untuk pemeriksaan akhir masing-masing part beda), dan waktu perjalanan ***bebas*** (simulasi 8 jam)

TUGAS PENDAHULUAN

PRAKTIKUM ARENA 2

TIPE 3

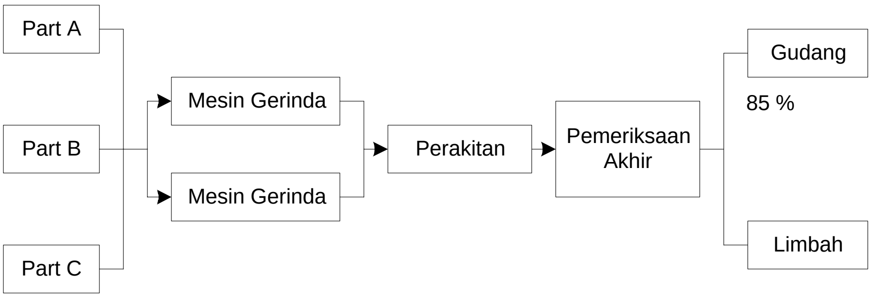
1. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Sequence
   - AdvServer
   - Queue
   - Set
   - Resource
   - Expression
2. Buatlah model dibawah ini dengan menggunakan software Arena !



Dengan data waktu antar kedatangan, waktu proses (khusus untuk pemeriksaan akhir masing-masing part beda), dan waktu perjalanan ***bebas*** (simulasi 8 jam).

TIPE 4

1. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Sequence
   - AdvServer
   - Queue
   - Set
   - Resource
   - Expression
2. Buatlah model dibawah ini dengan menggunakan software Arena !



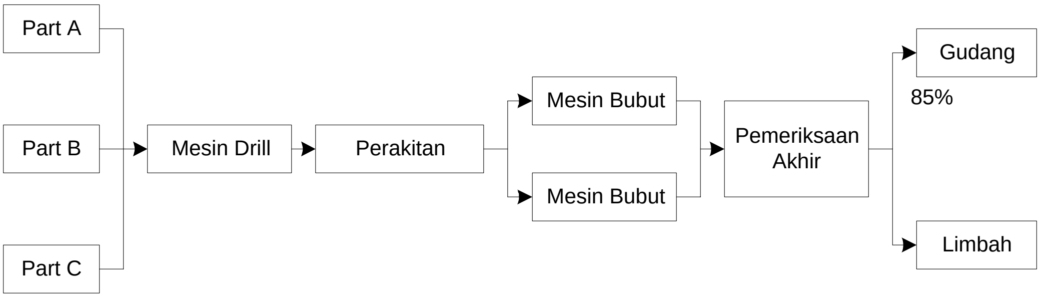
Dengan data waktu antar kedatangan, waktu proses (khusus untuk pemeriksaan akhir masing-masing part beda), dan waktu perjalanan ***bebas*** (simulasi 8 jam).

TUGAS PENDAHULUAN

PRAKTIKUM ARENA 2

TIPE 5

1. Jelaskan apa yang saudara ketahui tentang modul-modul dibawah ini :
   - Sequence
   - AdvServer
   - Queue
   - Set
   - Resource
   - Expression
2. Buatlah model dibawah ini dengan menggunakan software Arena !



Dengan data waktu antar kedatangan, waktu proses (khusus untuk pemeriksaan akhir masing-masing part beda), dan waktu perjalanan ***bebas*** (simulasi 8 jam).