Final Exam for *Computer Simulation* ***Spring 2005***

>>> SOLUTION <<<

Welcome to the comprehensive final exam in Computer Simulation. Read each problem carefully. There are ten required problems (each worth 10 points) and one extra credit problem worth 10 points. You may have a calculator and one 8.5 x 11 inch sheet of paper with you. On this sheet you may have anything you want (definitions, formulas, flow charts, etc.) in your handwriting on both sides of the page. You may not have photocopies or printed text on your formula sheet. Please answer each problem on a separate sheet of paper, unless otherwise noted. This exam includes a copy of Figure 2.5 (page 56) and Table 4.1 (page 98) from MacDougall.

Good luck!!! ☺

Problem #1

Answer the following general questions about performance and modeling (all in the context of this class):

a) What is performance?

Performance is the quantitative measure of a system.

b) What is a model? What is the purpose of a model?

A model is a representation (physical, logical, or functional) that mimics another object under study (Molloy, 1989). We build models to be able to make predictions about objects (e.g., to predict the performance of a system).

c) Why do we build models (as opposed to experiment on actual systems)?

It is frequently cheaper, safer, and faster to build and experiment on a model than on an actual system. In some cases an actual system may not exist and thus experimenting on a model is sometimes the only possible option.

d) Define computer simulation

Computer simulation is the discipline of designing a model of an actual or theoretical physical system, executing the model on a computer, and analyzing the execution output. (Fishwick, 1995)

e) What is the performance measure of most interest for information systems?

System delay or response time is of most interest.

Problem #2

Answer the following questions about probability theory:

a) Why is probability theory of interest to performance modeling of information systems?

Most events in information systems are characterized by randomness. Probability theory is the tool kit for understanding the behavior of such systems.

b) Define probability (either the experimental or axiomatic definition is sufficient).

The experimental definition of probability is $\Pr[outcome] = \lim\limits_{n \to \infty} \frac{\text{\# observations of an outcome}}{\text{n repetitions of experiment}}$.

The axiomatic definition is that for a sample space of U a probability measure P is a function of 1) P[U] = 1, 2) for any A contained in U, then P[A] greater than or equal to 0, and 3) for any A and B contained in U where A intersect B is the empty set, then P[A union B] = P[A] + P[B].

c) What is a random variable?

A random variable is a function that maps a real number to every possible outcome in the sample space.

d) Consider a random variable *X* which takes on values 1 and 2 with probability 0.25 and 0.75, respectively (i.e., $\Pr[x=1]=0.25$ and $\Pr[x=2]=0.75$). Determine the mean and variance of *X*. Plot the probability density function (pdf) and probability distribution function (PDF) of *X*.

$Mean = 0.25\cdot 1 + 0.75\cdot 2 = 1.75$. $Variance = \left(0.25\cdot 1^2 + 0.75\cdot 2^2\right) - 1.75^2 = 0.1875$. Plots of pdf and PDF are:

```
f(x) |                     F(x) |
1.00 +                     1.00 +       +---
     |                          |       |
0.75 +       +             0.75 +       |
     |       |                  |       |
0.50 +       |             0.50 +       |
     |       |                  |       |
0.25 +   +   |             0.25 +   +---+
     |   |   |                  |   |
     +---+---+---               +---+---+---
         1   2  x                   1   2  x
```

e) Write a C function that will return an empirically distributed random value corresponding to the random variable *X* in (d). You may assume that a function `randval()` that return uniform(0,1) exists.

```
double emp(void)
{
  if (randval() < 0.25)
    return(1.0);
  else
    return(2.0);
}
```

Problem #3

Answer the following questions about queueing theory:

a) Describe the Kendall notation for queues

A/S/c/k/m where A is arrival distribution, S is service distribution, c is number of servers, k is capacity of the system in customers, and m is numer of customers in the universe. Anything that is infinity is omitted. A and S can be M for Markov (Poisson), D for Deterministic, and so on.

b) State Little's Law

$L = \lambda W$ where L is queue length, λ is arrival rate, and W is delay in the system (wait).

c) If an M/M/1 queue has utilization of 80%, what is its mean queue length? If the arrival rate is 100 jobs per second (and utilization is 80%), what is the mean response time?

$L = \frac{\rho}{1-\rho}$ so L = 4. Using Little's Law we have that W = 40 milliseconds.

d) If an M/D/1 queue has utilization of 80% do you expect its mean queue length and response time to be less, same, or greater than than of an M/M/1? Explain your answer.

An M/D/1 has less variability than an M/M/1, hence the mean queue length and response time will be less than that of an M/M/1 for a given utilization.

Problem #4

Answer the following questions about discrete event simulation:

a) Sketch the basic flowchart for a discrete-event simulation model

See your notes. The top box is “initialize” and the bottom box is a decision box for “Done”.

b) Describe the concept of an event list (and explain how it works in the context of the flowchart in (a) above)

The event list is a linked list of event structures ordered by time of next event. An event structure consists of (at least) an event id and time to occur. Events are dequeued from the head of the event list (in the “Determine next event” box of the flowchart) and inserted into the list in the “Generate New Event” box of the flowchart.

Problem #5

Attachment #1 is an SMPL simulation model. What does this program model? Give the output from the execution of this program.

This program models a single server queue with deterministic interarrival and service times. The program output is:

```
(1) at 0.000000
(2) at 0.000000
(2a) at 0.000000
(1) at 1.000000
(2) at 1.000000
(3) at 1.200000
(2) at 1.200000
(2a) at 1.200000
(1) at 2.000000
(2) at 2.000000
(3) at 2.400000
(2) at 2.400000
(2a) at 2.400000
(1) at 3.000000
(2) at 3.000000
(3) at 3.600000
```

Problem #6

Attachment #2 is a CSIM simulation model. Give the output from the execution of this program.

The program output is:

```
(1) at 0.000000
(1b) at 0.000000
(1c) at 0.000000
(2) at 10.000000
(2) at 10.000000
(2) at 20.000000
END at 1000.000000
```

Problem #7 (write your answer on attachment #3)

Attachment #3 is a CSIM simulation model of a single server queue. Modify the program such that 1) if the number in the system is greater than 10 an arriving customer will not enter the queue with probability 0.50 (i.e., the probability of balking is 0.50 when the number in the system is 10 or greater) and 2) a served customer has probability 0.10 of immediately re-entering service (i.e., it does not re-queue) with the same service time as its just finished service.

See the attachment for the solution. New and modified code is highlighted in yellow.

Problem #8

You have simulated two systems for 7 replications each. The sample means for response time for system #1 are 100, 105, 110, 108, 102, 112, and 98 milliseconds. For system #2 they are 100, 105, 110, 108, 100, 110, and 95 milliseconds. Can you state with 95% and/or 90% confidence that one system is better (i.e., has lower response time) than the other system? Show your work.

We have D = 0, 0, 0, 0, 2, 2, and 3 milliseconds (the response time of system #2 subtracted from the response time of system #1). Thus, it appears that system #2 may be better (has lower response time). Computing:

$\overline{D} = \frac{1}{7}\sum_{i=1}^{7} D_i = 1.0$ and $S = \sqrt{\frac{1}{7-1}\sum_{i=1}^{7}\left(D_i - \overline{D}\right)^2} = 1.291$. For 90% confidence and 6 degrees of freedom the t score is 1.94, for 95% confidence it is 2.45. Thus, the half-width for 90% CI is $1.94 \cdot \left(1.291/\sqrt{7}\right) = 0.947$ and for 95% CI it is $2.45 \cdot \left(1.291/\sqrt{7}\right) = 1.195$. The mean difference ( ) plus/minus the 90% CI does not cross zero, but plus/minus the 95% CI it does cross zero. Thus, with only 90% confidence (and not 95% confidence) can we say that system #2 is better than system #1.

Problem #9

Answer the following questions regarding run length control and the simulation process:

a) Discuss the advantages and disadvantages of 1) independent replications and 2) batch means method for run length control.

Independent replications has the advantage of independence between runs and the disadvantage of many start-up transients to bias the results and/or force a longer overall run time. Batch means method has the advantage of only one start-up transient, but the disadvantage of correlation between batches.

b) Sketch the steps in a simulation study (i.e., the simulation flow chart we covered in class or the one in MacDougall). Describe the difference between model verification and validation.

See your notes for the flowchart. Verification = did you build it right (to the specification), validation = did you build the right thing (to requirements).

Problem #10

Answer the following questions regarding the mini case studies we did in class:

a) What was the purpose of the model (i.e., what question did we want to answer) in the RAID case study? What kind of model did we use?

The purpose of the model was to determine when the old RAID would saturate (i.e., mean response time exceed the stated maximum) and thus when a new, faster RAID should be bought and installed. A simple single-server queue model was used.

b) What method did we use to develop the model for the ATM (bank) case study?

We used successive refinement to “drill down” to the level of detail needed.

c) For the call center case study, what was the performance measure of interest? Give at least three ways that this performance measure can be improved (and that could be evaluated with our model)?

The performance measure of interest was percentage of calls successfully answered by an operator (or percentage of lost calls). The percentage of successful calls can be increased by training the operators to serve customers on less time, increasing the number of lines, ports, and operators, and even increasing the amount of time people will spend on hold by, say, playing better music.

Extra Credit

What are some of the current "open research problems" in the area of simulation?

Two open research problems that we discussed in class are 1) developing more realistic models of traffic (workload) that capture long range dependency (self-similarity) properties, and 2) developing sophisticated mathematical methods to allow for efficient simulation of rare events. Another problem of interest (that we discussed) is how to parallelize simulation models.

Attachment #1

```
#include <stdio.h>
#include "smpl.h"

void main(void)
{
  real Ta = 1;
  real Ts = 1.2;
  real te = 3.5;
  int customer = 1;
  int event;
  int server;

  smpl(0, "Mystery program");

  server=facility("server", 1);

  schedule(1, 0.0, customer);

  while (time() < te)
  {
    cause(&event,&customer);

    switch(event)
    {
      case 1:
        printf("(1) at %f \n", time());
        schedule(2, 0.0, customer);
        schedule(1, Ta, customer);
        break;

      case 2:
        printf("(2) at %f \n", time());
        if (request(server, customer, 0) == 0)
        {
          printf("(2a) at %f \n", time());
          schedule(3, Ts, customer);
        }
        break;

      case 3:
        printf("(3) at %f \n", time());
        release(server, customer);
        break;
    }
  }
}
```

Attachment #2

```
#include <stdio.h>
#include "csim.h"

STORE    Line;

void call(int alloc);

void sim(void)
{
  create("sim");

  Line = storage("line", 4);

  call(2);
  printf("(1) at %f \n", clock);
  call(2);
  printf("(1b) at %f \n", clock);
  call(2);
  printf("(1c) at %f \n", clock);

  hold(1000.0);
  printf("END at %f \n", clock);
}

void call(int alloc)
{
  int    status;
  double time_out;

  create("call");

  time_out = 100.0;
  status = timed_allocate(alloc, Line, time_out);
  if (status != TIMED_OUT)
  {
    hold(10.0);
    deallocate(1, Line);
    printf("(2) at %f \n", clock);
  }
  else
    printf("(3) at %f \n", clock);
}
```

Attachment #3

```
#include <stdio.h>
#include "csim.h"

FACILITY Server;

void generate(double lambda, double mu);
void queue1(double service_time);

void sim(void)
{
  double   lambda;
  double   mu;

  create("sim");

  Server = facility("Server");

  lambda = 1.0;
  mu = 3.0;

  generate(lambda, mu);
  hold(1000000.0);

  report();
}

void generate(double lambda, double mu)
{
  double   interarrival_time;
  double   service_time;

  create("generate");

  // Loop forever to create customers
  while(1)
  {
    interarrival_time = exponential(1.0 / lambda);
    hold(interarrival_time);

    service_time = exponential(1.0 / mu);
    queue1(service_time);
  }
}

void queue1(double service_time)
{
  create("queue1");

  if ((qlength(Server) + num_busy(Server)) >= 10)
    if (prob() < 0.50) return;

  reserve(Server);
  do
  {
    hold(service_time);
  } while (prob() < 0.10);
  release(Server);
}
```