Simulation Libraries and Random Variate Generation

Spring 2010

Dr. Louis Luangkesorn

University of Pittsburgh

January 26, 2010

Outline

1 Goals
2 Simulation Software
3 Types of Simulation
4 Software features
5 Example Sim101
   - Introduction to Sim101
Goals
Simulation software

A simulation should have the following components:

- System state: The collection of state variables which describe the system.
- Simulation clock: A variable giving the current value of simulated time.
- Event list: A list containing the next time when each type of event will occur.
- Statistical counters: Variables used for storing statistical information about system performance.
- Initialization routine: Initialize the simulation model at time zero.
- Timing routine: Determines the next event from the event list and then advances the simulation clock to that event.
- Event routine: Updates the system state when a particular event occurs.
- Library routines: Generate random observations from probability distributions.
- Report generator: Computes estimates of the desired measures of performance and produces a report when the simulation ends.
- Main program: Invokes the timing routine to determine the next event then transfers control to the corresponding event routine to update the system state.
Classifications of software

- HISTORICAL: Simulation languages
- General purpose languages
- Simulation packages

Simulation languages

- e.g. GPSS, SIMAN, SIMSCRIPT, SIMULA
- Provide most of the features needed in programming a model
- Basic building blocks provide a natural framework for simulation
- Usually have dynamic storage allocation for state variables
- Better error detection (because common errors are checked for automatically)
- Model tends to be more compact
Simulation Software

Simulation Software

Simulation packages

- e.g. ARENA, Extend, Anylogic, GoldSim, Simio, Witness, Promodel
- All the benefits from simulation languages. Often built on top of a simulation language
- Graphical model building - Pre-made components
- Fast development. Usable by those who are only occasional modelers.
- Only easy if it fits pre-made world view.

General Purpose Languages

- Procedural languages - e.g. C, Fortran
- Object Oriented - C++, Java, C# (?) (Note: The first OO language was SIMULA, which was created for the purposes of writing simulations)
- Dynamic languages - Visual Basic, Perl, Ruby
- Data languages - Matlab, R, Python, F# (?)
- Efficiently written programs tend to be faster
- More flexible
- More difficult to write (more details to keep track of)
- Simulation libraries may provide the advantages of simulation languages
Types of Simulation

Discrete event simulation approaches

- Event-scheduling - System is modeled by identifying its characteristic events then write event routines that give a description of how the system changes when the event occurs.
- Process approach - System is described by describing the experience of an entity as it goes through the system.
- Process approach tends to be more natural to describe.
- Event approach tends to be more flexible and easier to implement.

Software features

Desirable software features

- General purpose
- Animation
- Statistics
- Customer support
- Output reports
Software features

General purpose

- Modeling flexibility - The software should not force approximations in representing parts of the model.
- Ease of model development - Ease of specifying and debugging the model.
- Fast model execution speed - Because models can take a long time for a run.
- Maximum model size - Should not be forced by software, should be forced by hardware.
- Compatible across different types of computers - Desktop, parallel, super-computer.
- Cross simulation types - Discrete, continuous, monte-carlo, agent-based.

Animation

- In an animation, key elements of the simulation are represented visually.
- Can be concurrent, where it the animation is generated as the simulation is run.
- Can be run in playback, where the animation is displayed after the simulation based on a file with record of system state changes.
- Easily communicates the model.
- Some easy debugging.
- Can demonstrate that the model is NOT valid.
- Understand the dynamics of the system.
- Training personnel on system.
Animation disadvantages

- Does NOT replace statistical analysis. Essentially, an animation is a single sample.
- Only displays part of the model logic at a time.

Statistical capabilities

- Contain standard distributions
- Include *empirical* distributions
- Multiple-stream random number generators
- Run a series of independent replications
- *Warmup period* to begin statistical collection
- Construct confidence intervals on output statistics
Customer support

- Documentation and tutorials
- Articles and examples
- Conferences
- Seminars
- Technical support

Output reports

- Standard reports - Make it easy to get started.
- Tailored reports - Make it easy to customize a report.
- Graphics - Histograms, time series, box-plots
- Access to individual model output observations - In addition to summary statistics.
- Export to database, text file for analysis via statistical software.
Introduction to Sim101

- Written by Leemis and Lawson to go with *Leemis and Park, 2006, Discrete Event Simulation: A first course*.
- Used in multiple Wintersim Simulation 101 Workshop.
- Versions in C and R with same API.
- Suitable for Monte-Carlo and event-scheduling approaches.
- See paper for website to download either C or R versions.
- Intended for educational use NOT production use.

Libraries

- Rngs: a multiple-stream Lehmer random number generator.
- Rvgs: a library used to generate random variates from six discrete (Bernoulli, binomial, Equilikely, geometric, Pascal, and Poisson) and seven continuous (uniform, exponential, Erlang, normal, log-normal, chi-square, and student) distributions.
- Rvms: a library used to evaluate the probability density functions, cumulative distribution functions, and inverse distribution functions for the distributions provided in Rvgs.
Craps example

- craps
- Initialization
- Random variate generation
- System rules
- Collect statistics
- Return output

M/G/1 queue example

- ssq3
- Random variate generation
- System rules
- Next event list
- Statistics
- Graphical output