Chapter 3

Simulation Software

CONTENTS

3.1 INTRODUCTION

- Activities common to most simulations:
  - Random-number generation … draws from U(0, 1) distribution
  - Random-variate generation … draws from probability distributions specified as part of the inputs to the model
  - Advancing simulated time
  - Determining the next event from the event list, and passing control to the appropriate event logic
  - Adding records to lists, deleting records from lists
  - Collecting output statistics and reporting results
  - Detecting error conditions
- Simulation software packages are designed to do these things (and more) for you

3.2 COMPARISON OF SIMULATION PACKAGES WITH PROGRAMMING LANGUAGES

- Advantages of simulation packages
  - Provide most modeling features, so "programming" effort, cost is reduced, often significantly
  - Natural framework for simulation modeling
  - Usually make it easier to modify models
  - Better error detection for simulation-specific errors
- Advantages of general-purpose programming languages
  - More widely known, available
  - Usually executes faster … if well written
  - May allow more modeling flexibility
  - Software cost is usually lower

3.3 CLASSIFICATION OF SIMULATION SOFTWARE

- General-purpose vs. application-oriented packages
  - Traditionally: *simulation languages* and *simulators*
    - Languages were flexible but required programming, simulators were easy to use but not very flexible
  - Now, almost all simulation software uses graphical interface so is relatively easy to use, learn
  - Distinction now is between general-purpose simulation software and applications-oriented package
    - Specific applications include manufacturing, call centers, telecommunications, etc.

3.3 Classification of Simulation Software (cont'd.)

- Modeling approaches
  - *Event-scheduling* approach – as in Chaps. 1 and 2
    - Can uses general programming languages, or some simulation languages
    - During processing of an event, no simulated time passes
  - *Process-interaction* approach
    - Now used by most simulation software
    - Instead of identifying events, identify *entities* (a.k.a. *processes*) that are created, flow around or through the system, maybe leave
    - May have multiple realizations of an entity/process
    - May have different kinds of entities/processes
    - "Program" consists of a description of what happens to the different kinds of processes (including their entry and exit)
    - Usually expressed graphically, like a flowchart
    - During processing of an entity/process, simulated time usually passes

3.3 Classification of Simulation Software (cont'd.)

- Common modeling elements
  - *Entities* – represent customers, parts, messages, paperwork, airplane, etc.
  - *Attributes* – Information stored with each entity
    - Usually, every individual entity has the same set of attributes, but the values differ to distinguish the entities
    - Some attributes are automatic, others are user-defined and user-maintained
  - *Resources* – servers, machines, workers, nodes, links, runways, gates, agents, clerks, etc.
  - *Queues* – where entities wait if resources are not available

3.4 DESIRABLE SOFTWARE FEATURES

- General capabilities
  - Modeling flexibility – ability to drill down to lower levels of programming, create custom modeling constructs
  - Ease of use
  - Hierarchical modeling – submodels containing submodels, etc.
  - Fast execution speed
  - Ability to create user-friendly front/back ends for template creation
  - Run-time version for wide distribution of model
  - Import/export data from/to other applications
  - Automatic execution of models for different input-parameter combinations
  - Combined discrete/continuous modeling
  - Ability to initialize in other than empty & idle state
  - Save state at end to re-start later
  - Affordable

3.4 Desirable Software Features (cont'd.)

- Hardware and software requirements
  - Matches platform/OS – Windows, UNIX, MacOS
- Animation and dynamic graphics
  - Concurrent vs. postprocessing
  - 2D vs. 3D
  - Import CAD drawings
  - Display statistics, graphs dynamically during execution

3.4 Desirable Software Features (cont'd.)

- Statistical capabilities
  - Adequate random-number generator for basic U(0, 1) variates
    - Statistical properties, cycle length, adequate streams and substreams
    - RNG seeds should have good defaults, be fixed – not dependent on clock
  - Comprehensive list of input probability distributions
    - Continuous, discrete, empirical
  - Ability to make independent replications
  - Confidence-interval formation for output performance measures
  - Warmup
  - Experimental design
  - Optimum-seeking
- Customer support and documentation
- Output reports and graphics
  - Standard defaults, customizable – stored in database for postprocessing

3.5 GENERAL-PURPOSE SIMULATION PACKAGES

- See text for discussion of two popular general-purpose simulation packages – Arena and Extend
  - In each, builds a model of a small manufacturing system
- Mentions some additional general-purpose simulation packages
  - AweSim, Micro Saint, GPSS/SLX, SIMPLE++, SIMUL8, Taylor Enterprise Dynamics

3.6 OBJECT-ORIENTED SIMULATION

- OO programming and OO simulation originated in the same product – SIMULA, from the 1960s
- OO simulation has objects that interact as simulation progresses through simulated time
- Objects contain data, methods
- Also have encapsulation, inheritance, etc.
- Recent software product for OO simulation – MODSIM III

3.7 EXAMPLES OF APPLICATION-ORIENTED SIMULATION PACKAGES

- Oriented toward specific classes of applications – see book for software packages for:
  - Manufacturing
  - Communications
  - Process reengineering and service systems
  - Health care
  - Call centers
  - Standalone animation – links to multiple simulation-modeling packages