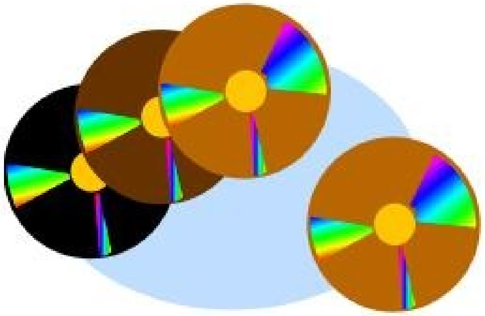
Simulation Software

Discrete-Event System Simulation

5th Edition

Chapter 4



World Views of Simulation Model

- Event-Scheduling View
  - As with our project 1
  - Focus on processing each event
- Process-interaction View
  - View model as a set of processes through which an entity “flows”
  - Life-cycle approach – time-sequenced list of events, activities, & delays
  - Common in simulation environments

World Views of Simulation Model

- Activity Scanning Approach
  - Focus on activities & conditions that allow it to begin
  - At each clock advance, scan conditions to start any activity that can begin
  - Approach is simple, but scan is slow
  - New 3-phase approach includes some event scheduling – somewhat more complex but more efficient

Categories of Simulation Software

- General Purpose Languages
  - C, C++, Java
- Simulation Languages
  - GPSS, SIMAN, SLAM, SSF
- Simulation Environments
  - Enterprise Dynamics, Arena, SIMUL8

Features of Simulation Languages

- Some focus on a single type of application
- Built in features include
  - Statistics collection
  - Time management
  - Queue management
  - Event generation

Features of Simulation Environments

- Some focus on one type of application
- Icon based
- Analysis of I/O
- Advanced Statistics
- Optimization
- Support for Experimentation

History of Simulation Software

(Nance 1995)

1955-60	Period of Search
1961-65	Advent
1966-70	Formative Period
1971-78	Expansive Period
1979-86	Period of Consolidation & Regeneration
1987- 2008	Period of Integrated Environments
2009 +	The Future

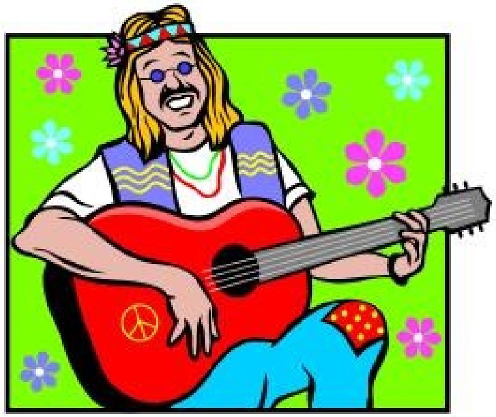
Simulation Languages

- 1981 – 137 Simulation languages reported
- More have be developed since
- Now Simulation Environments

The Search:: 1955 - 60

- FORTRAN – one of a few languages
- Focus on unifying concepts & reusable functions
- General Simulation Program – first effort at “language” which as a set of functions

The Advent:: 1961-65



- GPSS – 1961 @ IBM
  - Based on block diagrams
  - Well-suited for queuing models
  - Expensive at first
- SIMSCRIPT – 1963 – Rand Corp.
  - US Air Force – government is biggest user
  - FORTRAN influence
  - Owned by CACI in CA.

The Advent:: 1961-65

(continued)

- GASP – 1961
  - Based on Algol, then Fortran
  - Collection of Fortran functions
- SIMULA – extension of Algol
  - Widely used in Europe
- CSL (Control & Simulation Language)

Formative Period:: 1966-70

- Concepts caused major revisions of languages
- Languages gained wider usage
- GPSS (several variations)
- Simscript II – English-like
- ECSL – Europe
- SIMULA – added classes & inheritance

The Expansion Period:: 1971-78

- GPSS/H – 1977
- GASP IV – 1974 – Purdue
- SIMULA
    - Attempt to simplify the modeling process
    - Program generators – severe limitations

Consolidation & Regeneration:: 1979-1986

- Movement to mini and PC computers
- SLAM II (descendant of GASP)
  - 3 world views
    - Event, Network, Continuous
- SIMAN (descendant of GASP)
  - General Modeling + Block Diagrams
  - 1st first major language - PC & MS-DOS
  - Fortran functions w/ Fortran programming

Integrated Environments:: 1987 - 2008

- Growth on PC's
- Simulation Environments
  - GUI
  - Animation
  - Data analyzers

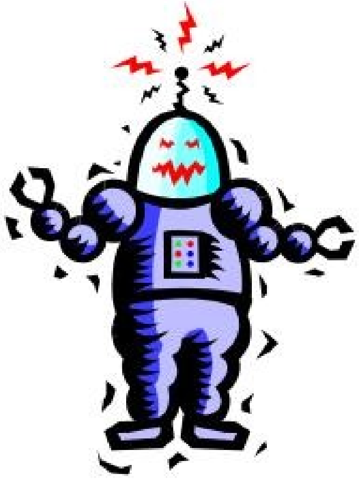
The Future :: 2009 - 2011

What can we expect in the future? (2008)

- Virtual Reality
- Improved Interfaces
- Better Animation
- Agent-based Modeling

Agent-Based Software

- AnyLogic
- Ascape
- MASON
- NetLogo
- StarLogo
- Swarm
- RePast



Evaluating Software

- Consider multiple issues
  - Ease of use, support, applicability
- Speed of execution
  - Experimental runs – Debugging
- Beware of demos & advertising
  - Will focus on strengths only
  - Ask for demo of YOUR problem

Evaluating Software

- Carefully consider comparison checklists with yes/no answers
- Can software link to external languages
- Carefully consider trade-off between graphical model building & simulation programming language
- Costs – one-time vs. licensing

Simulation Software Features

See the following tables in text:

- Model-building features
  - P. 123 – Table 4.1
- Runtime Environment
  - P. 124 – Table 4.2
- Animation & Layout features
  - P. 124 – Table 4.3

Simulation Software Features

- Output features
  - P. 125 – Table 4.4
- Vendor Support - Documentation
  - P. 125 – Table 4.5

Example Simulation

Checkout Counter – Single Server Queue

Consider at standard checkout counter environment with on clerk and one queue. Interarrival times are exponentially distributed with mean 4.5 minutes; service times normally distributed with mean 3.2 and standard deviation 0.6 minutes.

Simulate for 1000 customers.

Java Model

- Section 4.4 – p.126
- Note similarity to our process in project one

GPSS

General Purpose Simulation System

- Highly Structured
- Process Approach
- Queuing Systems
- Block Diagrams
  - 40 standard blocks
  - Block corresponds to a statement
- Transactions FLOW through the system

GPSS Block Diagram for Example

- Figure 4.10 – p. 138
- Each entity has a name
  - Name each queue, server, etc.
- In rectangle, parameters (as necessary)
- Right attachment, name of entity
- Far right column – GPSS Command

GPSS Syntax

Assembly-like

Label   OpCode Subfields ; comment

- Label: col. 1, <= 9 alphanumeric, alpha start
- OpCode: 4+ characters of command
- Subfields: as necessary, separated by commas
- Comment: after ; or with * in column 1

GPSS Program

- Figure 4.11 – p. 139
- Declaration Section
- Customized vs. Standard Output
- Code Section

```
      Generate     rvexpo (1,&IAT)
      Queue        Systime
      Queue        Line
      Seize        Checkout
      Depart       Line
      Advance      rvnorm(1,&mean,&stdev)
      Release      Checkout
      Depart       Systime
      Test_GE      M1, 4, Term
      Blet         &Count = &Count +1
Ter   Terminate    1
      Start        &Limit
```

GPSS Output

- Customized
  - Figure 4.12 – P. 141
- Standard
  - Figure 4.13 – P. 142

Other Simulation Software

- SSF – Scalable Simulation Framework
  - Application Program Interface (API)
  - Object-oriented, process view
  - 5 Base Classes
    - Process, Entity, Event, InChannel, OutChannel
  - Designed for high-performance computers
  - Bridges pure Java & simulation languages
  - Figures 4.14 & 4.15

Simulation Environments ~~ Common Features

- GUI
- Animation
- Automatic statistics
- Output (tables, graphs, custom)
- Analysis
- Process world view

Common Features (# 2)

- Some allow
  - Event Scheduling
  - Mixed continuous-discrete models
- Animations – 2D & 3D
- Business Graphics

Simulation Environments

- AnyLogic
- Arena
- AutoMod
- Enterprise Dynamics
- ExtendSim
- Flexsim
- ProModel
- SIMUL8

AnyLogic

- Supports: discrete event, agent-based, system dynamics (& combination)
- Hybrid: discrete & continuous
- Object library
- Java models, publish as applets
- Animation, Statistics, optimization, debugger

Arena

- Discrete & Continuous systems
- Object-based; GUI
- 2D, 3D Animation
- Business & Manufacturing processes
- Supports Analysis
- OptQuest for optimization
- Based on SIMAN; embedded Visual Basic

AutoMod

- Manufacturing & Materials handling
- Detailed large models for planning, decision support, control systems
- AutoStat - Experimentation & analysis
- AutoView - Make movies of 3D animations
- Full simulation language included

ENTERPRISE DYNAMICS®

- Object oriented
- Discrete Events
- Open GL 3D visualization engine
- 4D Script programming language
- Interfaces with databases
- OptQuest optimization

ExtendSim

- Block-diagram approach
- Versions for mixed and for continuous only
- Includes C-like programming language
- Supports linking to external languages

Flexsim

- Dynamic-flow systems - manufacturing
- Discrete-event, Object-oriented simulator; developed in C++ using Open GL
- Animation: 2D, 3D, Virtual reality
- Drag & Drop

ProModel

- Manufacturing Systems
- Simulation & Animation (2D & 3D)
- Output viewer – graphs, tables
- SimRunner – optimizer based on evolutionary algorithm technique
- OptQuest is also available
- MedModel, ServiceModel

SIMUL8

- Service industries, transaction processing
- Drop & Drag model development
- Saves in XML format
- Pre-built templates for common applications
- 3D virtual reality graphics
- Links to database

Experimentation & Statistical Analysis Tools

- Included in most all simulation systems
- Add-ons also available
- Features
  - Optimization – define fitness or cost function

Arena
Output & Process Analyzer

- Confidence intervals
- Comparison of systems
- Warm-up determinations
- Graphs (all types) – 2D & 3D
- Scenario definition

AutoStat (from AutoMod)

- Warm-up determination
- Steady state determination
- Confidence intervals
- Sensitivity analysis
- Optimization via evolutionary strategy

OptQuest

- Based on scatter search, tabu search, linear-integer programming, data mining, neural nets (evolutionary)
- Uncertainty problems
- Global optimums
- Handles non-linear and discontinuous relationships

SimRunner (from ProModel)

- Based evolutionary models & genetic algorithms
- Optimizations
- 3D graphics
- Warm-up (steady state) determination

Conclusion

- Many simulation software environments available
- Many do have trial versions to download for trying
- Before deciding, consider the features and the add-ons available that will suit your particular environment