Introduction to Simulation

Rosaldo Rossetti
rossetti@fe.up.pt
Http://www.fe.up.pt/~rossetti
Assistant Professor at FEUP – Faculty of Engineering of the University of Porto
Researcher at LIACC – Artificial Intelligence and Computer Science Lab.

Luís Paulo Reis
lpreis@fe.up.pt
Http://www.fe.up.pt/~lpreis
Assistant Professor at FEUP – Faculty of Engineering of the University of Porto
Member of the Directive Board of LIACC – Artificial Intelligence and Computer Science Lab.

Simulation

• Simulation is the imitation of some real thing, state of affairs, or process, over time, representing certain key characteristics or behaviours of the selected physical or abstract system.

• Simulation:
  – Modeling of natural systems or human systems in order to gain insight into their functioning.
  – Simulation of technology for performance optimization, safety engineering, testing, training and education.
  – Widely used tool for decision making, what if analysis.

• Applied to complex systems that are impossible to solve mathematically.
What is simulation?

- The imitation of the operation of a real-world process or system over time...
  - Most widely used tool (along LP) for decision making
  - Usually on a computer with appropriate software
  - An analysis (descriptive) tool – can answer what-if questions
  - A synthesis (prescriptive) tool – if complemented by other tools
- Applied to complex systems that are impossible to solve mathematically

A Few Examples of Applications

- Games
- Film Industry
- Manufacturing
- Bank operations
- Airport and Airlines
- Flight Simulation
- Military Operations
- Transportation
- Satellite Navigation
- Robotics
- Biomechanics
- Molecular Dynamics
- Logistics, supply chain, distribution
- Hospitals: Emergency, operation, admissions...
- Computer networks
- Business processes
- Chemical plants
- Fast-food restaurants
- Supermarkets
- Stock Exchange
- Theme parks
- Emergency-response systems
- Sports
A Few Examples of Applications

- War gaming
- Flight Simulators
- Transportation systems
- Aerodynamics: Wind Tunnel
- Robotics
- Games & Sports

System

- A set of interacting components or entities operating together to achieve a common goals or objectives

Examples

- A manufacturing system with its machine centres, inventories, conveyor belts, production schedule, items produced.
- A telecommunication system with its messages, communication network and infrastructure, servers.
- A theme park with rides, workers, ...
Metrics & Performance Measures

• The *Performance Metric* is a measurable quantity that precisely captures what we want to measure (e.g. response time, throughput, delay, etc.)

• For example, in computer systems, we might evaluate
  – The response time of a processor to execute a given task
  – The execution time of two programs in a multi-processor machine

• In Network systems, we might evaluate
  – The (maximum/average) delay experienced by a voice packet to reach the destination
  – The throughput of the network
  – The required bandwidth to avoid congestion

Metrics & Performance Measures

• The performance of a system is dramatically affected by the *Workload*

• The Workload characterises the quantity and the nature of the system inputs
  – For Web Servers, system inputs are http requests (GET or POST) The workload characterises
    • the intensity of the requests: how many requests are received by the web server. High intensities deteriorate the performance.
    • The nature of the requests: the request can be simple GET request or a request that require the access of a remote database. The performance will be different for different request types.
  – *Benchmarks*: used to generate loads that is intended to mimic a typical user behaviour.
Why & How to study a system?
Measure/estimate performance, improve operation/training, be prepared (for failures)

SYSTEM

Experiment with the Actual System
Too costly or disruptive
Not appropriate for the design

Experiment with a Model of the System
There is always the question of whether it actually reflects the system.

Mathematical Model
Make assumptions that take the form of mathematical or logical relationships

Analytical Solution
If the model is simple enough. E.g., calculus, algebra, probability theory

Simulation
Highly complex systems

Systems Modelling
• An abstract and simplified representation of a system
• Specifies
  – Important components
  – Assumptions/approximations about how the system works
• Not an exact re-creation of the original system!
• If model is simple enough, study it with Queuing Theory, Linear Programming, Differential Equations...
• If model is complex, Simulation is the only way!
### Systems Modelling

Models of the System

\[ I = \frac{E}{R} \quad F = IR \]
\[ R = \frac{E}{I} \quad P = EI \]
\[ h_{de} = \frac{I_c}{I_b} \quad h_{le} = \frac{I_c}{I_{le}} \]

### Variables of a Model

- **Input parameters**: uncontrollable/exogenous
- **Output parameters**: endogenous

- **Input parameters**: controllable/exogenous

Model
Getting answers from models

Operating Policies
- Single queue, parallel servers
- FIFO

Input Parameters
- No of servers
- Inter-arrival Time Distribution
- Service Time Distributions

Y = f(X)

Output Parameters
- Waiting Times
- System Size
- Utilizations

Stochastic Models

- Uncertainty (randomness) is an inherent characteristic
- Example: bank with customers and tellers

Actual System

Queuing Model
Classification of simulation models

1. **Static vs. Dynamic Models:**
   - **Static Models:**
     - Represents a system as it evolves over time (e.g., a conveyor system in a factory).
     - Time plays no role; represents a system at a particular point in time (e.g., Monte-Carlo methods).
   - **Dynamic Models:**
     - Involves random variables, probabilities (e.g., most queueing and inventory systems).

2. **Deterministic vs. Stochastic Models:**
   - **Deterministic Models:**
     - No probabilistic components (e.g., worst-case analysis of a system).
   - **Stochastic Models:**
     - Involves random variables, probabilities (e.g., most queueing and inventory systems).

3. **Continuous vs. Discrete Models:**
   - **Continuous Models:**
     - The state of the system changes only at discrete points in time.
     - The state of the system changes continuously (e.g., chemical processes).
   - **Discrete Models:**
     - Bit Arrival in a Queue
     - # of cars in a parking lot
     - Continuous Model
     - Discrete Model
Simulation Approaches

- Types of discrete models
  - Event-oriented
  - Process-oriented
  - Activity-oriented

How to simulate

- By hand
  - Buffon needle and cross experiments
    [http://www.ms.uky.edu/~mai/java/stat/buff.html](http://www.ms.uky.edu/~mai/java/stat/buff.html)
- Spreadsheets
- Programming in a general purpose language
  - C++, Java, C#
- Simulation languages
  - SIMAN, Simscript
- Simulation packages
  - Arena, Simulink

**Issue: modelling flexibility vs. ease of use**
Simulation is not Appropriate if?

- Problem can be solved by:
  - Common sense
  - Simple calculations
  - Analytical methods
  - Direct experiments
- Simulation costs exceed savings
- Resources & time are not available
- Data is not available
- Verification & validation are not practical due to limited resources
- System behavior is too complex (essential model is not easy to capture)?

Simulation Advantages

- Advantages of Simulation:
  - When mathematical analysis methods are not available, simulation may be the only investigation tool
  - When mathematical analysis methods are available, but are so complex that simulation may provide a simpler solution
  - Provides practical feedback when designing real world systems
  - Time compression or expansion
  - Higher Control
  - Lower costs
  - Comparison of alternative designs or alternative operating policies
  - Sensitivity Analysis
  - Training tool
  - Doesn’t disturb real system
Disadvantages of Simulation

- For a stochastic model, simulation estimates the output while an analytical solution, if available, produces the exact output.
- Often expensive and time consuming to develop.
- An invalid model may result with confidence in wrong results.
- Is quite dependent on available data and validation methods.

Life-cycle of a Simulation Project

1. Problem formulation
2. Setting of objectives and overall project plan
3. Data collection
4. Model conceptualization
5. Model translation
6. Verification
7. Validation
8. Implementation
9. Documentation and reporting
Life-cycle of a Simulation Project

1. **Problem Formulation**
   - Statement of the problem

2. **Set Objectives & Project Plan**
   - Questions to be answered
   - Is simulation appropriate?
   - Methods, alternatives
   - Allocation of resources

3. **Model Conceptualization**
   - Requires experience
   - Begin simple and add complexity
   - Capture essence of system
   - Involve the user

4. **Data Collection**
   - Time consuming, begin early
   - Determine what is to be collected

5. **Model translation**
   - Computer form
   - General purpose vs. special purpose lang.

6. **Verification**
   - Does the program represent model and run properly? Common sense

7. **Validation and Calibration**
   - Compare model to actual system
   - Does model replicate system?
   - How to calibrate the model?

8. **Experimental Design**
   - Determine alternatives to simulate
   - Time, initializations, etc.

9. **Production & Analysis**
   - Actual runs + Analysis of results
   - Determine performance measures

10. **More Runs?**
Problem formulation

- **A statement of the problem**
  - the problem is clearly understood by the simulation analyst
  - the formulation is clearly understood by the client
  - All involved elements and their characteristics, as well as behaviours and interactions are well identified

Setting of objectives & project plan

- **Project proposal**
  - Determine the questions that are to be answered
  - Identify scenarios to be investigated
  - Decision criteria
  - Determine the end-user
  - Determine data requirements
  - Determine hardware, software, & personnel requirements
  - Prepare a time plan
  - Cost plan and billing procedure
Model Conceptualisation

- Real World System
- Assumed system
- Conceptual model
- Logical model

Conceptual Model

- Abstract essential features
  - Events, activities, entities, attributes, resources, variables, and their relationships
  - Performance measures
  - Data requirements

- Select correct level of details (assumptions)
Level of Details

- Low levels of detail may result in lost of information and goals cannot be accomplished

- High levels of detail require:
  - more time and effort
  - longer simulation runs
  - more likely to contain errors
  - More available data for validation and verification

Accuracy of the model vs. Scope & level of details

Cost of model vs. Scope & level of details
Components of a System

- **Entity**: is an object of interest in the system
  - *Dynamic objects* – get created, move around, change status, affect and are affected by other entities, leave (maybe)
  - Usually have multiple *realizations* floating around
  - Can have different types of entities concurrently

**Example: Health Centre**

- Physicians?
- Patients
- Visitors

Components of a System

- **Attribute**: is a characteristic of all entities, but with a specific value "local" to the entity that can differ from one entity to another

**Example: Patient**

- Type of illness,
- Age,
- Gender,
- Temperature,
- Blood Pressure
Components of a System

• Resources: ...what entities compete for!
  – 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 which can be changed
    during the simulation

Example: Health Centre
Doctors, Nurses
X-Ray Equipment

Components of a System

• Variable: A piece of information that reflects some characteristic
  of the whole system, not of specific entities
  – Entities may access and change some variables
  – Other variables are changed as result of system dynamics

Example: Health Centre
Number of patients in the system,
Number of idle doctors,
Current time
Components of a System

- **State**: A collection of variables that contains all the information necessary to describe the system at any time

  **Example: Health Center**

  - Number of patients in the system,
  - Status of doctors (busy or idle),
  - Number of idle doctors,
  - Status of Lab equipment, etc...

Components of a System

- **Event**: An instantaneous occurrence that changes the state of the system

  **Example: Health Centre**

  - Arrival of a new patient,
  - Completion of service (i.e., examination)
  - Failure of medical equipment, etc...
Components of a System

- **Activity**: represents a time period of specified length

**Example: Health Centre**

- Surgery,
- Checking temperature,
- X-Ray.

Logical Model

- Shows the logical relationships among the elements of the model (flowchart)

- L : # of entities in system
- Q : # of entities in queue
- B : # of entities in server

1. Departure event
2. L(t) = L(t-1)
3. B(t) = B(t)
4. Q(t) = Q(t) - 1
5. Generate service & schedule new departure
6. Collect & update statistics TB, TQ, TL, N

L(t)=L(t)-1
Q(t)=Q(t)-1

YES

NO
Data Collection & Analysis

- Collect data for input analysis and validation
- May require the help of subjects (e.g. Measures, survey forms such as SP)
- Analysis of the data
  - Determine the random variables
  - Fit distribution functions

Model Translation

- Simulation model executes the logic contained in the flow-chart model

Coding

- General Purpose Language
  - Examples: JAVA, C++, C#, Python
- Special Purpose Simulation Language/Software
  - Examples: SIMAN, ARENA, EXTEND
Model Translation

- **Visual Interactive Modelling & Simulation**

- **IDE example: Arena**

```java
public static void main(String argv[])
{
    Initialization();
    //Loop until first "TotalCustomers" have departed
    while (NumberOfDepartures < TotalCustomers)
    {
        Event evt = FutureEventList[0]; //get imminent event
        removeFromFEL(); //be rid of it
        Clock = evt.get_time(); //advance in time
        if (evt.get_type() == arrival) ProcessArrival();
        else ProcessDeparture();
    }
    ReportGeneration();
}
```
Verification & Validation

- **Verification**: the process of determining if the operational logic is correct.
  - Debugging the simulation software

- **Validation**: the process of determining if the model accurately represents the system.
  - Comparison of model results with collected data from the real system
Experimental Design

- Alternative scenarios to be simulated
  - *Calibration* is necessary to guarantee the scenario is accurate
- Type of output data analysis
  (steady-state vs. terminating simulation analysis)
- Number of simulation runs
- Length of each run
- The manner of initialization (warm-ups)
- Variance reduction

Result Analysis

- Statistical tests for significance and ranking
  - Point Estimation
  - Confidence-Interval Estimation
- Interpretation of results
- More runs?
Documentation & Reporting

• **Program Documentation**
  – Allows future modifications
  – Creates confidence

• **Progress Reports**
  – Frequent reports (e.g. monthly) are suggested
  – Alternative scenarios
  – Performance measures or criteria used
  – Results of experiments
  – Recommendations

Implementation

• Deployment of results...