COURSE DESCRIPTION

Dept., Number  CSC 148  Course Title  Modeling and Experimental Design
Semester hours  3  Course Coordinator  William Mitchell
URL (if any):  http://gaia.ecs.csus.edu/~mitchell/

Catalog Description

Modeling and simulation techniques in system representation. Problem analysis, model formulation, data collection and analysis, experimental design, testing, verification and validation, and simulation experiments. Monte Carlo methods. Queuing theory. Term projects. Prerequisites: At least a C- grade in both MATH 31 and STAT 50, proficiency in at least one programming language, and full CSC or MATH/CSC major status.

Textbook


References

Course Website: http://gaia.ecs.csus.edu/~mitchell/csc148.

Course Goals

1. To apply appropriate techniques in modeling and simulation.
2. To study various modeling and simulation tools used to analyze and design systems.
3. Learn one discrete event simulation language.
4. Apply elementary experimental design techniques.

Prerequisites by Topic

Thorough understanding of:
• At least one high-level programming language.
• Data structures.
• Trace and debugging skills used in typical programming language environments.

Basic understanding of:
• Integral and differential calculus.
• Probability and statistics.
• Algorithm analysis.
• Interpreted/compiled languages.
Major Topics Covered in the Course

1. Modeling, simulation, simulation examples (3 hours).
2. Statistical models in simulation, random number generation, random variate generation (5 hours).
3. Programming discrete-event simulations including simulation-language features (GPSS currently used as the course simulation programming language) (8 hours).
4. Simulation languages, and in-depth introduction to runtime system and structure of simulation languages; simulation scheduling algorithms and object representation (6 hours).
5. Input modeling: collecting sample data and assimilating it (by distribution fitting) into a model and simulation; significance testing with chi square and F distributions (6 hours).
6. Queuing models and Kendall classifications (9 hours).
7. Multivariate and time-series models (2 hours).
8. Output modeling and analysis of variance (ANOVA); experimental design (4 hours).
9. In-class quiz, bi-weekly, throughout the semester (2 hours).

Laboratory Projects

1. Modify a simple single server queue model; analyze report output (1 week).
2. Learn the programming environment for the simulation language GPSS; editing source, declarations, simulation block types and control (1.5 weeks).
3. Implement a simple inventory model with orders, reorder and stock out processing, or alternatively a manufacturing/assembly line model (2 weeks).
4. Utilize trace, debug and visualization tools for simulation systems (1 week).
5. Perform data collection for arrival, queuing and service aspects of an existing system such as a bank ATM or fast food drive through; carry out fitting analysis to test the collected data against a theoretical distribution (3 weeks).
6. Compare steady state queuing model formulas against simulation results for certain Kendall classification queuing systems (2 weeks).
7. Perform ANOVA studies for a simulation of a network of queues; user-coded event scheduling; simulating server/entity failure and recovery (3 weeks).

Estimated Curriculum Category Content (Semester hours)

<table>
<thead>
<tr>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
<th>Area</th>
<th>Core</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithms</td>
<td></td>
<td>1.0</td>
<td>Data Structures</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software Design</td>
<td></td>
<td></td>
<td>Prog. Languages</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>Comp. Arch.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Oral and Written Communications

The distribution-fitting and ANOVA assignments require written analysis and description in appropriate technical writing format (3-4 written pages). Use of a UML modeling tool requires written analysis and design prose documentation for design of event, state, and state transitions simulation logic. (Construction of state transition diagrams and embedded documentation “notes.”) In addition, programming-assignment source code documentation is critiqued (and counted as part of the assignment grade for each simulation program) throughout the semester.

Social and Ethical Issues

No significant component.

Theoretical Content

1. Queuing theory and steady state equations (2 weeks).
2. Distribution fitting (3 weeks).
4. Multivariate and time-series input models (1 week).

Problem Analysis

Students are required to analyze structure and behavior aspects of real systems. Default and customized output are also identified (since simulators involve both). Structure includes boundaries, interfaces and physical characteristics of the target system. Structures such as users, machines, and waiting lines can be represented by narratives as well as informal visual notations. Fundamental behavior usually includes events, delays, communication and state transitions (and/or payload flows depending on the problem domain). Behaviors are represented using narratives as well as UML sequence, and state diagrams (as appropriate).

Solution Design

Mathematical distribution and probability modeling is employed to simulate system randomness. The fundamental constructs identified in a simulation design include flows and rates, arrivals/departures, queued services, event lists and object interactions. An analysis behavior such as a delay becomes a design construct such as a bounded FIFO queue. Because simulation practice often deals with multiple runs with controlled/varied inputs, design also includes the formal experimental design for factor treatments and ANOVA. For small-scale systems, design might lead to an implementation of the target system by manual simulation/tracing. In some cases, design and solution can take a purely mathematical formulation. When neither tracing nor mathematical solutions are feasible, a design must translate into discrete-event software simulation.
In the discrete simulation case, design must also address model validation, an additional aspect of software development not common to other kinds of software. In addition, simulation design includes simulation run control for initializing, suspending, resuming, and terminating (the possibly numerous) runs. This is often needed for multivariate analysis and is always needed for generating simulation model contours and response surfaces.