Introduction & building simulations

Simulation and Modeling (CSCI 3010U)

Faisal Qureshi

Part I
Why do we care about simulations?

Simulation

A model that you can manipulate to answer a question

Simulation

- A computational model of some phenomena
- More than a set of equations, or the solution of an equation at a particular point
- Ability to try out different scenarios, look at different situations, ask "what if" questions

Example: Car Simulation

Applications

- Car design (model-based design)
- Road design
- Driver education
- Entertainment

Things we can simulate...

- Physics of car motion
- Terrain or ground
- Controls and displays

Benefits

- Evaluate car designs under different driving conditions
- Easy to carry out many design iterations

Example: Military Simulation

Scenario

- US Army uses tank simulations for tank designs
- Simulated tanks are pitched agains each other in simulated battles comprising thousands of soldiers
- Designs that survive go to the prototype stage

Assumptions about simulation

- Tank simulations relied upon data gather during the first Gulf war, which had a lot of tank-vs-tank battles
- Second Gulf war didn't see much tank-vs-tank action; gorilla warfare, road-side bombs, etc.
- The tanks that were designed for tank-vs-tank battle didn't perform well under these conditions

Key Lesson

- Simulations are only as good as their input data
- One needs to be *very careful* in defining the scope of the problem
  - Consider all possible cases
  - Simulations can be incorrect even if there are no *mathematical* or *programming* errors
- The military simulations above did not take into account road-side bombs, which resulted in tanks that were not equipped to deal with such situations

Why do we build simulations?

- Validate a model or a theory
- Perform experiments that are too expensive or too dangerous to carry out
- Training
- Model-based design
- Decision making and problem solving
- Education

Validate a model or a theory

- Need some evidence that our theory is correct
- Does it behave the way we think it should?
- Does it mimic the real world?
- Where is it inaccurate?

Performing experiments

- Simulations can be cheaper than real experiments, and give the same results
- What will happen in the case of severe weather? Don't want to actually destroy buildings or kill people!
- Control of nuclear reactors, pollution of water bodies, global warming, etc.

Model-based design

- Producing prototypes is expensive, eliminate poor designs quickly using simulation
- Fine tune designs, test different parameter combinations quickly
- Try the design in real situations to see how it will perform

Decision making and problem solving

- Even with a mathematical model the optimal solution may not be obvious
- May not be able to mathematically solve the problem
- What is the best number of bank tellers under particular conditions?
- How should an assembly line be configured?

Education

- Show how something works, rather than talk about it
- Students can explore the model, try different things
- Active learning, see how the simulation responds, try theories and ideas
- Can build the simulation themselves

Training

- Real equipment is expensive and dangerous
- Replicate training situations, such as bad weather and emergency situations
- Instructor can provide better feedback, pause the simulation and explain the problem
- More access to simulator than the real thing
- No ability to train on real thing, space program

Part II
Building simulations

Building simulations

- Requirements
- Modeling
- Data Collection
- Implementation
- Validation
- Use

Requirements

- Why are we building the simulation?
- How will it be used?
- We need to know what the problem is before we can start building a model and implementing the simulation
- The people who want the simulation can tell us how it will be used, what they need it for

Requirement gathering

- Problem statement
- Background material, standards, etc.
- Accuracy of the simulation
- Time requirements, how fast must the simulation run?
- Important parameters
- Output

Modeling

- We need to develop a model of the phenomena that we are simulating
  - How the system changes over time
  - How does it react to different events
- Two main classes of models
  - Continuous systems simulations
  - Discrete event systems simulations
- Continuous and DES simulations can be combined to construct hybrid simulations

Continuous systems simulations

- View time as a continuous variable
- Differential equations describe how the *state* evolves over time
- Inputs are parameters, which can be controlled by the user

Example: car simulator

- Variables: speed, heading, amount of fuel, etc.
- Inputs: steering wheel position, force on gas and brake pedals, road conditions, etc.
- Model: differential equations that take into account the forces (engine thrust, ground friction, etc.) acting on the car

Discrete event systems simulations

- These focus on *events*
- Everything is based on the occurrence of events, which occur at discrete points in time
- An event often leads to more events in the future, and so on
- Events can begin to accumulate at some object, forming queues, and output is usually the average time the system takes to process an event
- Quite often involve statistics, probability distributions and queuing theory
- Can be used to model computer systems and networks

Example: ATMs

- Customers entering a bank
- What percentage of these customers use an ATM
- What do they use ATM for: withdrawls or deposit
- How long do they spend at the ATM
- When do these customers leave the ATM

Data collection

- Most models will have parameters and input data
- Need to collect this information before you can implement or run the model
- Sometimes the information is easy to collect, such as different physical constants, just a matter of looking it up in a book
- Other types of data won't be readily available, average time for a teller to process a customer
- You may need to set up an experiment to collect the data, monitor bank tellers for several days and collect statistics

When data collection becomes part of the simulation results

- In some cases you may even need to guess, the data becomes part of the simulation problem, what values give reasonable results?

Consistency in data collection

Keep an eye on units

- With physical constants they must all use the same set of units, if not you will need to do conversions
- Equations won't work if the constants have different units, they will produce incorrect values
- This is a major source of errors

Keep an eye on the process used to collect data

- In some cases if the data isn't collected in the same location it may not fit together, must be from same population
- In the bank example, all statistics should be captured in the same bank. Different banks could process customers in different ways, so mixing the statistics could produce meaningless results.
- This could be one of the hardest problems in simulation development

Implementation

Basically producing a computer program for the simulation

Continuous system simulations

- In the case of continuous system simulations this usually involves writing the program for the simulation
- We will use the open source physics framework that provides most of the program code
- We still need to convert the differential equations into program code, build the interface to the simulation

Discrete event simulations

- Discrete event simulations are a bit easier, standard programs exists for performing the simulation
- Must still describe the system to be simulated, the events and the objects that process the events
- Programs produce standard statistics
- May still need to program if you need something special

Implementation considerations

- Implementation effort depends upon simulation application
- For some applications implementation is relatively easy, put something together quickly a let it run
- Often the case if the results aren't time sensitive, can afford to wait for the simulation to finish

Time critical simulations

- In other cases time is important
- For training simulators you must have real time response, must respond to user input
- For weather predictions, must have tomorrow's weather before tomorrow arrives
- Some simulations could take years to run if they aren't implemented correctly

Implementation

We will look at some of the implementation issues later in the course:

- Use of multiple processors to reduce time
- Handling large amounts of data in large simulations
- Networking and remote simulations

Validation

- Before we can use a simulation we need to validate it, determine whether it is giving the correct values
- If the simulation is wrong, we don't want to use the results
- There are several ways that we can do this, the easiest is to compare the results to known values

Validation

- If there is a special case where the output is known, then check this case
- Run the simulation in parallel with the real world
- Use known inputs, the current situation and see if the simulation predicts what actually happens
- Example: input today's weather conditions, and see if tomorrow's weather prediction is correct
- This only works if you can capture the required input

Validation

- See if the simulation predicts known behavior
- If you know that a particular condition arises under certain circumstance, then see if the simulation produces similar results
- Not the best validation, but shows whether the simulation is going in the right direction

Use

- In some cases the simulation will only be used a few times to answer a particular set of questions
- In other cases it will be used on a daily basis, in this case may need to update the simulation as conditions change

Example

- A bank wants to improve service times, but doesn't know whether it should add tellers or ATMs
- Best answer may vary from bank to bank depending on customer mix, current facilities, etc
- Can use a simulation to help answer this question

Example

- Really want to have satisfied customers, but this isn't something a simulation will tell us
- If they are serviced quickly, they are likely to be happy, and we can measure service time, so that will be our output
- Need to develop a model for this simulation

Example

- Customers arrive at the bank, want to perform one or more transactions
- Customer arrival isn't continuous, arrive at discrete times -> discrete event simulation
- Model based on transactions, time to perform these transactions at teller and ATM
- Also model customer choice between teller and ATM

Example

There will be a distribution of:

- Customer arrival time
- Number of transactions
- Type of transactions
- Preference for teller or ATM
- We will need to measure this information, could vary from one bank to another
- Also need to know the current number of tellers and ATMs

Example

- Can use a standard discrete event simulation package for this problem
- Can verify the simulation by running it for several days using current bank - configuration and statistics
- Check if simulated waiting times agree with the measured waiting times

Example

- Run the simulation with different numbers of tellers and ATMs
- Will need to run the simulation several times in order to obtain accurate results
- May also want to vary the inputs to see how sensitive the results are, does a small decrease in customer arrival time produce a much longer waiting time?

Summary

- Examined the process of creating simulations
- Notice the parallels and the important differences with software engineering
- Identified two main types of simulations
  - Continuous systems simulation
  - Discrete event simulations
- Now we can start looking at the details of designing and implementing simulations, starting with continuous systems