CS 308-522A: Modelling and Simulation

M&S for Software Engineering for M&S . . .
Overview

1. What is Modelling and Simulation?
2. Which topics does CS522 cover?
3. How to get an A?
4. What are the assignments about?
5. Python?
6. Where do I get the material covered in CS522?
7. Multiple formalisms?
What is Modelling and Simulation?

Why?

- Modelling: represent/re-use/exchange knowledge about system structure and behaviour
- Simulation: to emulate real behaviour
  - cost, danger, ...
  - what-if
  - optimization (do it right the first time)
Modelling and Simulation . . .

- . . . is Systems Theory, Control Theory
- . . . is Numerical Analysis, Computer Algebra
- . . . is Computer Science, Artificial Intelligence
- . . . is Operations Research
- . . . is Application Domain: Mechanical Engineering, . . .

. . . or more GENERIC ?
Modelling and Simulation Process

REALITY

- Real-World entity
- System S
- Experiment Observed Data
  - only study behaviour in experimental context
  - experiment within context

MODEL

- Base Model
- Model M
- Simulation Results
  - simulate = virtual experiment
  - validation

GOALS

- Model Base a-priori knowledge
  - Modelling and Simulation Process
Which topics does the course cover?

1. Rapid Application Development with Python.
2. The Modelling and Simulation Process + Block Diagram example.
5. Untimed Discrete Event Formalisms:
   (a) (non)Deterministic State Automata.
   (b) Adding Concurrency and Synchronisation: Petri Nets (cfr. UML, specifying network protocols).
   (c) Adding Hierarchy and Orthogonality: State Charts (cfr. UML, specifying reactive software).
6. Timed Discrete Event Formalisms:
(a) Timed Automata.
(b) Event Scheduling (*e.g.*, parallel simulation of networks).
(c) Activity Scanning (AI).
(d) Three Phase Approach.
(e) Process Interaction (GPSS).
(f) DEVS as a rigorous basis for hierarchical modelling.

7. Deterministic Simulation of Stochastic Processes:
   (a) Pseudo Random Number Generation.
   (b) Gathering Statistics (performance metrics).

8. Continuous Time Formalisms:
   (a) Ordinary Differential Equations & Algebraic Equations.
   (b) Differential Algebraic Equations.
   (c) CSSLs: sorting and algebraic loop detection.
   (d) Forrester System Dynamics, Population Dynamics.
(e) Hybrid (continuous-discrete) modelling and simulation.

How to get an A?

- 30% on 5 (out of 6) highest scoring small assignments.
- 30% on 3 implementation assignments.
- 15% on the project.
- 25% on the final exam.
What are the small assignments about?

1. Petri Net model for intersection/roundabout
2. State Chart model for CD player GUI
3. Process Interaction model for intersection/roundabout
4. GPSS assignments with simulator (+ ATOM3 GUI)
5. Modelica electrical system
What are the assignments involving coding about?

(GUI behaviour should be modelled with State Charts)


2. DEVS simulation of intersection/roundabout (2 styles) + performance metrics + animation.

What are the project subjects?

- GUI automatically generated from FSA specs
- Petri Net deadlock analysis
- Petri Net simulation environment with model animation.
- ES, AS simulators applied to cashier/queue.
- Process Interaction simulator applied to GPSS examples.
- Hybrid simulator applied to bouncing ball problem.
- Computer algebra: causality assignment and sorting.
Assignment rules of the game?

- Completely on WWW: requirements, design, code, discussion.
- All coding in Python.
- In groups of max. 3 people (alone is feasible).
- Original work, present in class.
- Respect deadlines or do more work to compensate.
Python?

- Why?


- Tkinter tutorial this Friday at 16:45 in MC320.
Python scripting: Why?

- **purpose:** glueing, system integration, *complimentary* to system programming languages.

- **examples:** command shells (sh, csh, zsh, ...), Perl, Tcl, Python, Visual Basic, ...

- **Rapid Application Development (RAD):**
  - interpreted (no edit/compile/link/test/...)
  - weak typing, but verbose and safe error diagnostics
  - garbage collected
  - powerful basic structures (control, dictionaries, exceptions, ...)
  - classes are first class objects, introspection

- **glue through extensions**
• embedding in large applications for user extension
• incremental development: dynamic loading (.so, .dll)
• platform-independent (Mac, Windows, UNIX)
• plethora of existing extensions (numerical, graphical (OpenGL, Tkinter), networks, . . . )
Where do I get the material covered in CS522?

- Course pack covering the whole course.
- Class presentations online in PDF format.
- Links and references for those interested.
Multiple formalisms? Car suspension
Multiple Formalisms: Physical
Multiple Formalisms: Mechanical

\[ \begin{align*}
\text{M} & \quad \text{m} \\
F_2 & \quad \text{F}_3 \\
K & \quad D \\
m & \quad \text{mg} \\
k & \quad \text{mg} \\
\text{road} & \\
0 & \quad x_0
\end{align*} \]
Multiple Formalisms: Electrical

\[ \text{Diagram with electrical components: } \begin{align*} &v_0 \\ &v_1 \\ &v_2 \\ &k \\ &F_1 \\ &F_2 \\ &F_3 \\ &D \\ &M \\ &F_m \end{align*} \]
Multiple Formalisms: Transfer Function
Multiple Formalisms: Bond Graph
Multiple Formalisms: ODE

\[ v_k = v_0 - v_1 \]
\[ F_1 = \frac{1}{k} \int v_k dt \]
\[ F_m = F_1 - F_2 - F_3 \]
\[ v_1 = \frac{1}{m} \int F_m dt \]
\[ v_D = v_1 - v_2 \]
\[ F_3 = \frac{v_D}{D} \]
\[ v_K = v_1 - v_2 \]
\[ F_2 = \frac{1}{K} \int v_K dt \]
\[ F_M = F_2 + F_3 \]
\[ v_2 = \frac{1}{M} \int F_M dt \]
Multi-formalism modelling

Paper Pulp mill

Waste Water Treatment Plant

Fish Farm

System of WWTP and Stormwater tanks (DEVS)

Input/Output function

algae

fish

GERRA

CFA

CFF

EDRF+

GF

CS522: Modelling and Simulation
Process Interaction DEV: GPSS

SIMULATE

*  
*  
GPSS/H Block Section (the model)

*  
*  
MANUFACTURING SHOP - MODEL 1
*  
Time unit = 1 minute

*  
GENERATE  5 Create parts
ADVANCE  4,3 Inspect
TRANSFER  .1,ACC,REJ Select rejects
ACC TERMINATE  1 Accepted parts
REJ TERMINATE  1 Rejected parts

*  
*  
GPSS/H Control Statements (the experiment(s))

*  
START  1000 Run 1000 parts

END
Population Dynamics, System Dynamics

2-species predator–prey system
Trajectory