CPE 345
Modeling and Simulation

Lecture 1

Course Introduction

- Instructor: Cristina Comaniciu
  - Office: Burchard 211
  - Phone: 216-5606
  - E-mail: ccomanic@stevens.edu
  - Office hours: Monday 1.30 pm – 3.00 pm, or by appointment
  - For all information needs to be sent to the class, the pipeline e-mail list for the class will be used
    - Please make sure you check your e-mail for the address listed there
  - Course website:
    - http://www.ece.stevens-tech.edu/~ccomanic/cpe345_05.html - available soon

- Course requirements and grading
  - Homework: 20% - individual effort
  - Midterm 30% - individual effort
  - Final 30% - individual effort
  - Project 20% - group effort – small groups 2-4 students encouraged

Course requirements

- Homework:
  - Generally due one week after it was assigned
  - Problem solutions will be posted on line – no late homework will be accepted after the solution was posted
  - Must be printed hardcopy, or electronic (not handwritten)
    - If electronic submission by e–mail attachment:
      - The file should explicitly contain your name
      - No executable attachments or macroviruses
  - Include the problem statement with your solution
  - Keep a copy of your homework – it will probably not be returned
  - For programming submissions, make sure you include all the required files (e.g. .h, initialization files, etc.) so that the program can be built and run
  - Try to solve the homework correctly and concisely (limit your results to a few pages)

Textbook and software

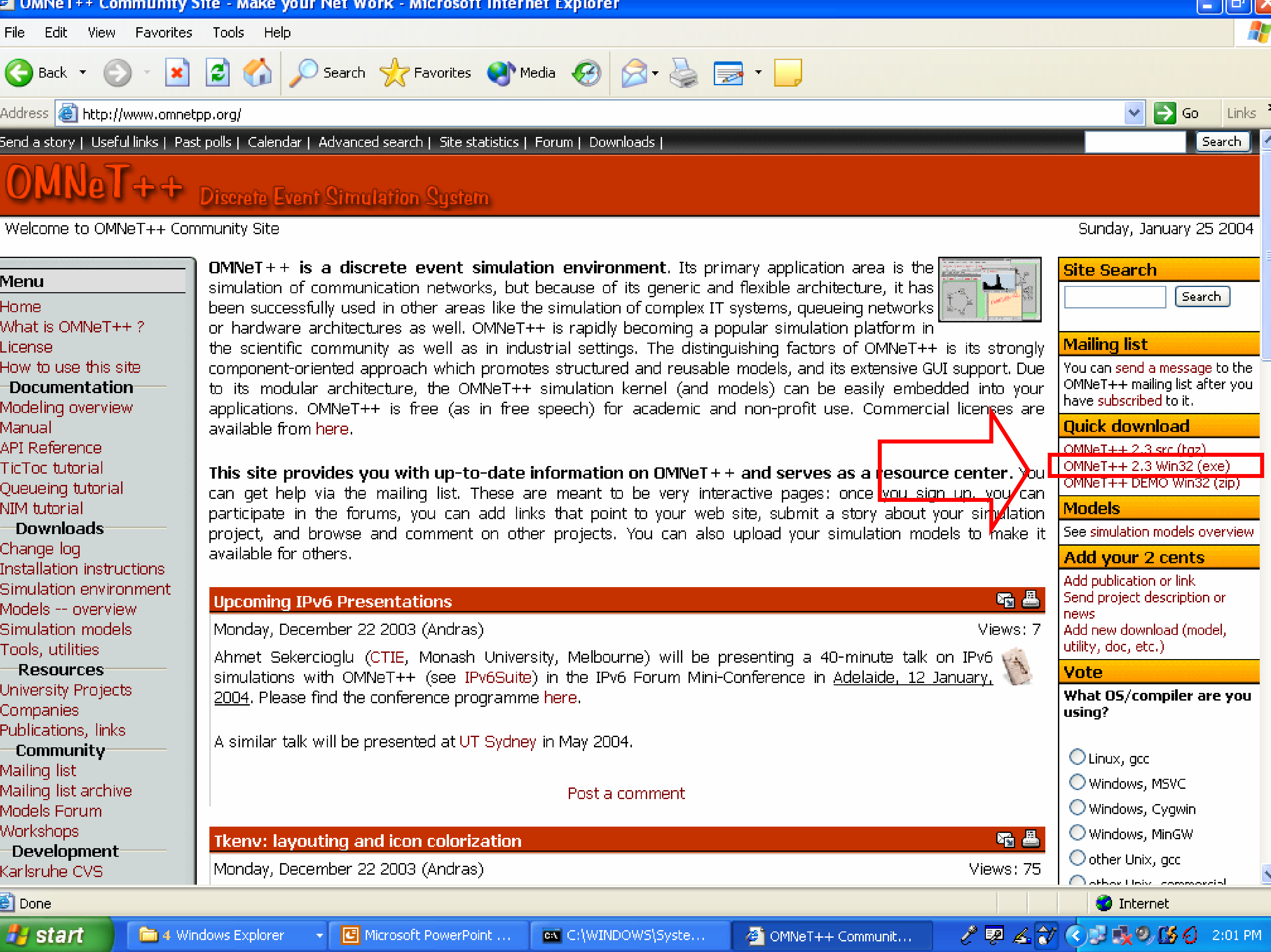
- Textbook:
  - Banks, Carson, Nelson & Nicol, “Discrete Event System Simulation, Prentice Hall, 2001.
  - Supplemental: OMNET++ user manual, available on line, as a part of the free software package:

    http://www.omnetpp.org/

- Software: OMNET++: event driven simulator - required
  - Free software for academic use, download from http://www.omnetpp.org/, use the quick download link on the right hand side of the web page (currently version 3.0)
    - Download from pipeline web page for class, version 2.3.
  - MS. Visual C++ 6.0: you can obtain a copy from Stevens which has a license for undergraduates. Installation of OMNET++, requires that you have previously installed MS. Visual C++ 6.0.
  - Matlab may also be used occasionally to plot various results – not required – examples may be presented in class, that are generated with Matlab
    - Free similar program: Euler Software: - not personally tested

      http://mathsrv.ku-eichstaett.de/MGF/homes/grothman/euler/index.html
    - Matlab compatible free software: Octave – not personally tested

      http://www.octave.org


OMNeT++ Community Site - Make your Net Work - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.omnetpp.org/
Go Links
Send a story | Useful links | Past polls | Calendar | Advanced search | Site statistics | Forum | Downloads |
Search
OMNeT++ Discrete Event Simulation System
Welcome to OMNeT++ Community Site
Sunday, January 25 2004
Menu
Home
What is OMNeT++ ?
License
How to use this site
Documentation
Modeling overview
Manual
API Reference
TicToc tutorial
Queueing tutorial
NIM tutorial
Downloads
Change log
Installation instructions
Simulation environment
Models -- overview
Simulation models
Tools, utilities
Resources
University Projects
Companies
Publications, links
Community
Mailing list
Mailing list archive
Models Forum
Workshops
Development
Karlsruhe CVS
OMNeT++ is a discrete event simulation environment. Its primary application area is the simulation of communication networks, but because of its generic and flexible architecture, it has been successfully used in other areas like the simulation of complex IT systems, queueing networks or hardware architectures as well. OMNeT++ is rapidly becoming a popular simulation platform in the scientific community as well as in industrial settings. The distinguishing factors of OMNeT++ is its strongly component-oriented approach which promotes structured and reusable models, and its extensive GUI support. Due to its modular architecture, the OMNeT++ simulation kernel (and models) can be easily embedded into your applications. OMNeT++ is free (as in free speech) for academic and non-profit use. Commercial licenses are available from here.
This site provides you with up-to-date information on OMNeT++ and serves as a resource center. You can get help via the mailing list. These are meant to be very interactive pages: once you sign up, you can participate in the forums, you can add links that point to your web site, submit a story about your simulation project, and browse and comment on other projects. You can also upload your simulation models to make it available for others.
Upcoming IPv6 Presentations
Monday, December 22 2003 (Andras)
Views: 7
Ahmet Sekercioglu (CTIE, Monash University, Melbourne) will be presenting a 40-minute talk on IPv6 simulations with OMNeT++ (see IPv6Suite) in the IPv6 Forum Mini-Conference in Adelaide, 12 January, 2004. Please find the conference programme here.
A similar talk will be presented at UT Sydney in May 2004.
Post a comment
Tkenv: layouting and icon colorization
Monday, December 22 2003 (Andras)
Views: 75
Site Search
Search
Mailing list
You can send a message to the OMNeT++ mailing list after you have subscribed to it.
Quick download
OMNeT++ 2.3 src (tgz)
OMNeT++ 2.3 Win32 (exe)
OMNeT++ DEMO Win32 (zip)
Models
See simulation models overview
Add your 2 cents
Add publication or link
Send project description or news
Add new download (model, utility, doc, etc.)
Vote
What OS/compiler are you using?
Linux, gcc
Windows, MSVC
Windows, Cygwin
Windows, MinGW
other Unix, gcc
Done
Internet
start
4 Windows Explorer
Microsoft PowerPoint ...
C:\WINDOWS\Syste...
OMNeT++ Communit...
2:01 PM

Project and OMNET++ simulator

- The project will be an example of an event driven simulation and will be implemented using OMNET++ (probably starting from the examples that are already implemented)
- OMNET++ comes with a set of examples, a tutorial and a manual.
  - Please install the program, and try to compile and run the token ring example: token
  - Class examples: related to computer networks: queueing
  - Homework problems in OMNET++ will help you build the basic blocks for your project simulation
  - Some observations
    - Make sure you have already installed on your computer MS Visual C++ 6.0, before installing OMNET++
    - For compilation in Windows systems, refer to Chapter 8.3 in the user manual

Windows users: Compiling OMNET++ applications from the command line

- Click start button on your computer screen
- Chose run – a small window will open: type cmd and click with the mouse on OK, a command line window will open
- Change the directory to the current work directory (e.g. C:\OMNET++\samples\token)

1. Use the OMNET++ utility to create Makefile.vc
   - opp_nmakemake –f (will consider all files in the current directory)
2. Add dependencies nmake –f Makefile.vc depend
3. Compile nmake –f Makefile.vc
   - An executable program will be created which has the name of the directory (e.g. nim)

- Every time you make changes to the files, first type

  nmake –f Makefile.vc clean, than repeat from 3
- If you add files in the directory, you have to rerun 1, to create another Makefile

Course topics

- Modeling of physical systems
  - What are models, when to use them versus when to measure or analyze
- Simulation
  - What it is, when it is doable, when it makes sense
- Event driven simulation
  - General principles, and an example of simulator: OMNET++
  - Examples for Queueing systems and Communications/Computer systems
- Queueing models
- Random numbers
  - Uses, generation, validation, pitfalls
- Output analysis

Course outcomes

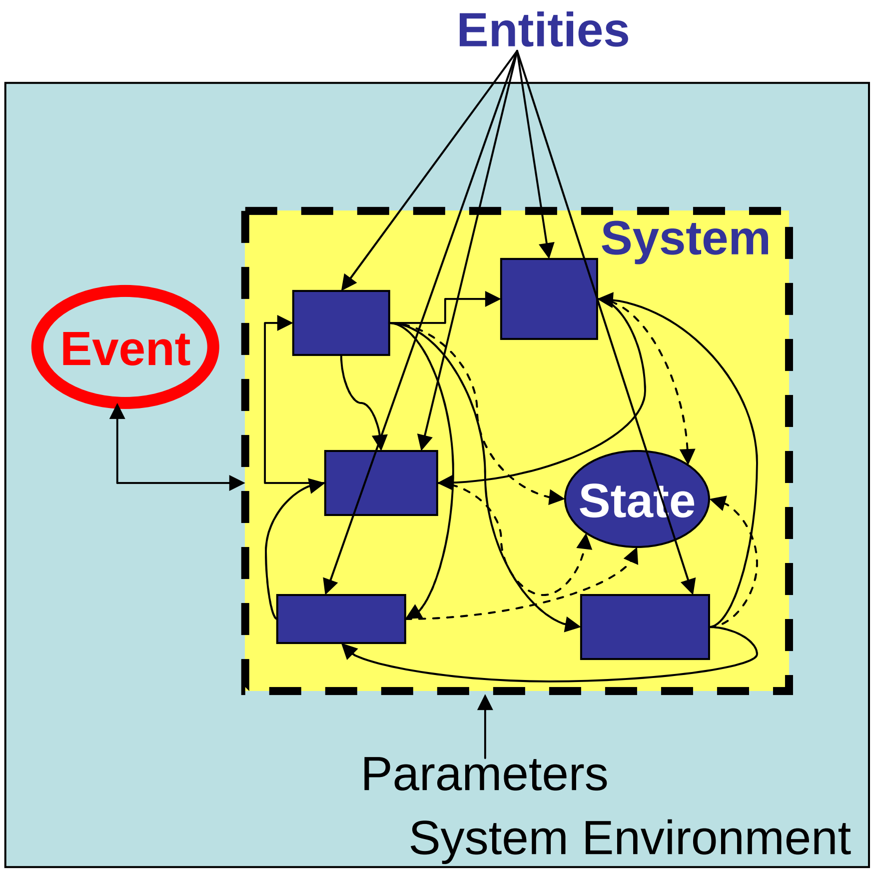
- ABET Criteria:
  - 1A1 – Recognize mathematical parameters as if they were physical variables and vice-versa
  - 1A2 – Be able to follow general mathematical concepts of derivation of engineering or scientific result and possess the mathematical skill to link those concepts
  - 1A3 – Be able to understand the relevance of the mathematical results to physical applications
  - 1A4 – Be able to articulate algorithmic thinking through flow charts
  - 3B1 – Use software for preparing, transmitting, and displaying multimedia documents
  - 3B2 – Have the ability to use computational tools for finding graphical, numerical, statistical and analytic solutions to problems
  - 3B3 – Have the ability to use systems simulations appropriate to engineering practice
  - 4A2 - Be able to identify input, output, and operating variables as appropriate in various units
  - 4A3 – Be able to identify technical relationships between the input, output and variables and use the relationships to predict mutual changes

Introduction to simulation

- What is simulation?
  - Imitation of the operation of a real-world process or system over time.
  - Generates an artificial history of a system and based on the observation of that artificial history inferences concerning the operating characteristics of the real system can be drawn.
  - A simulation can be only as good as the *simulation model* is.
  - A simulation model – set of assumptions concerning the operation of the system, expressed as mathematical, logical and symbolic expressions between the object of interests (entities) of the system.
  - From the simulation, data are collected as if a real system were being observed.
  - The simulation generated data – used to estimate the *measures of performance* for the system.

What is a system?

- System: set of objects, joined to accomplish some purpose



Entity – object of interest in the system

Attribute: property of an entity

Activity: predefined set of actions in a specified time period

State of system: collection of variables that describes the system at any time

Event: Instantaneous occurrence that may be associated with change of system state

System Environment: region outside the system that influences system behavior.
How to choose the boundary?

Simulation Model

- A simulation model will consider a system model and an environment model instead of the actual physical system
- What is a system model?
  - Abstraction of a real system
    - Simplifying assumptions capture only important behaviors, may even make analysis tractable in some cases
    - Outputs of the model will represent an estimate of the real outputs for the physical system
- In the simulation, time scale may be also altered as needed
  - Capture rare events – compress time
  - Simulate rapidly occurring events – expand time

When is simulation appropriate?

- Allows access to system internals that may otherwise not be observable.
- Informational, organizational, and environmental changes can be simulated, and the effect of these changes on the model's behavior can be observed.
- Observations based on simulations give great insight into the system behavior, and it can be determined which variables are most important and how they interact.
- Analytic solutions can be verified.
- Simulation allows to experiment with new designs or policies prior to implementation.
- Can be used for training without the cost and disruption of on-the-job learning.
- Animation shows a system in simulated operation so that the plan can be visualized
- The simulated system is so complex, that its interactions can be treated only through simulation

When simulation is not appropriate?

- Would common sense suffice?
- Is there an analytical solution?
- Is it easier to perform direct measurements on a physical system?
- Is the cost exceeding savings?
- Is there a shortage of resources for implementing the simulation (funds are lacking to purchase simulation tools, designing and conducting the simulation)?
- Is there a shortage of time for getting the desired results?
- Is data lacking for modeling the system and beginning a simulation study?
- Is there enough time and personnel to verify and validate the model?
- Are the managers' expectations unrealistic?
- Is the system to complex to be modeled?

Advantages and Disadvantages of Simulation

Advantages

- Explore new design options without disrupting existing systems
- Test new hardware, transportation systems, etc, without investing resources for their acquisition
- Time scale can be compressed (for slow moving systems) or expanded (for fast moving systems)
- Internal variables can be made observable
- Sensitivity and interaction of variables can be studied to understand their impact on the system behavior
- Bottleneck analysis can be performed
- Deployment options can be studied: what-if questions may be answered

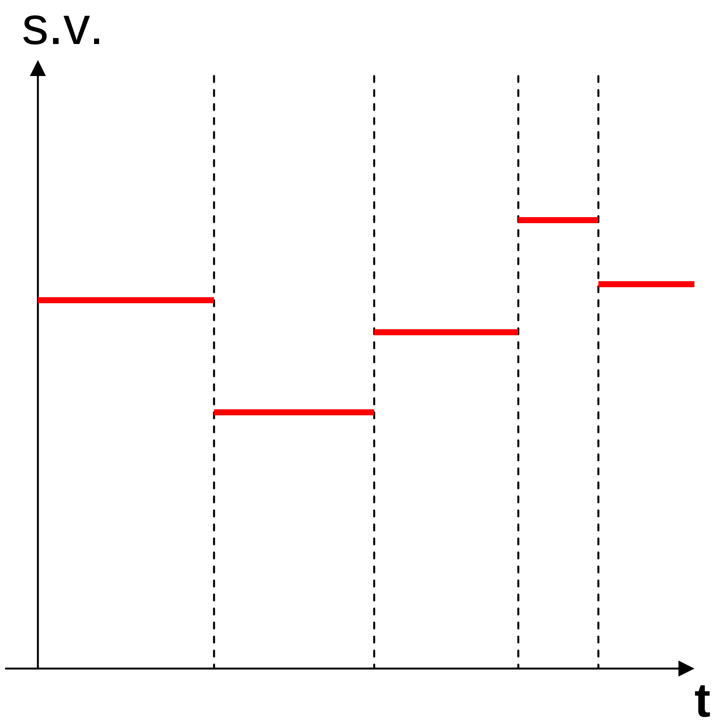
Disadvantages

- Model building may require special training
- Simulation packages may be expensive
- Learning curve to use simulation packages may be longer than the time available
- Closed form analysis may be possible
- Results may be difficult to interpret:
  - Do they really represent the system, or a small set of instances?
  - Is an observation characteristic for the system inter-relationships, or is just the result of randomness?
    - Usually simulation results are obtained for random inputs

Discrete versus Continuous System

Discrete Systems

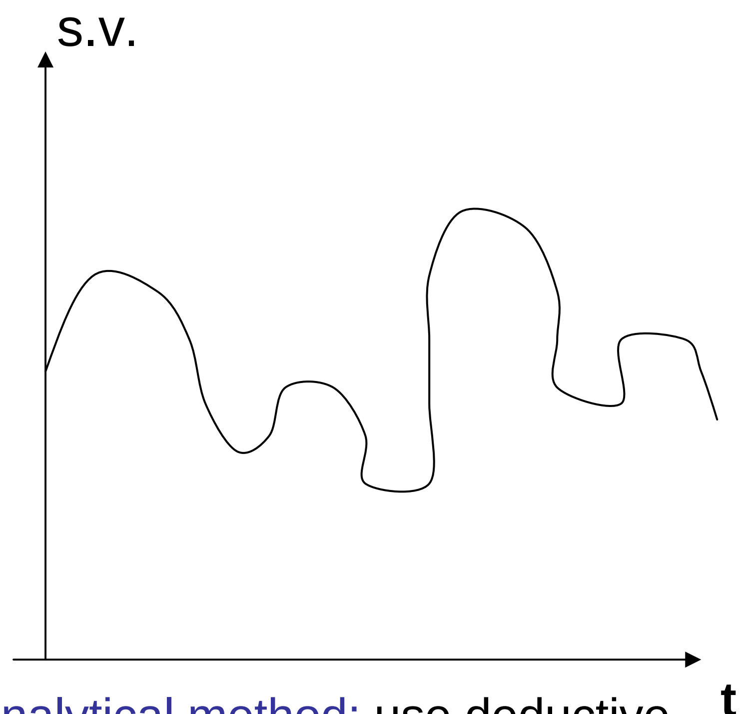
- State variable changes at discrete points in time (events)



Numerical method: use computational procedures to solve mathematical models

Continuous Systems

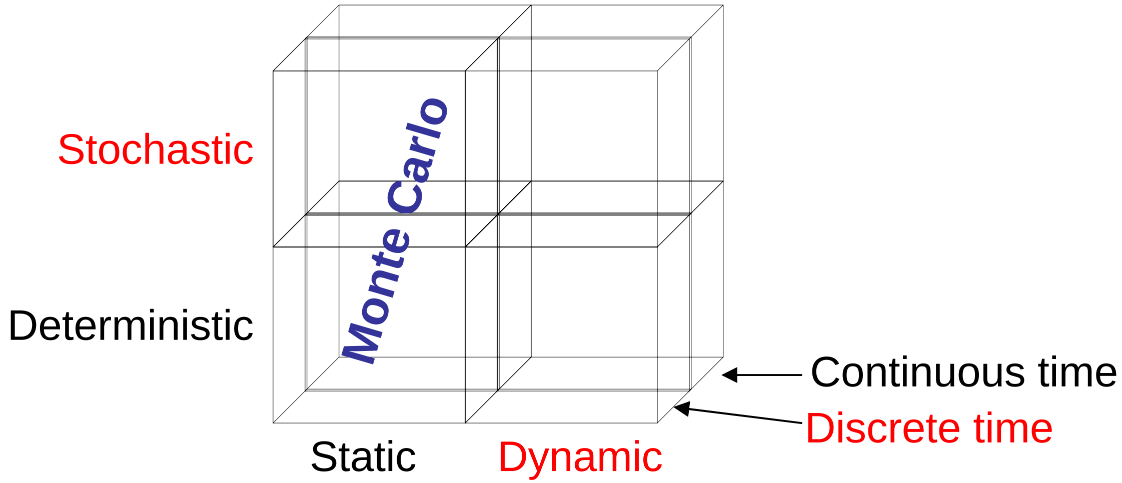
- State variable changes continuously, as a function of time



Analytical method: use deductive mathematical reasoning to define system and solve

Types of Models

- A simulation model – a particular type of mathematical model of a system. A mathematical model uses symbolic notations and mathematical equations to represent the system.
- General classification of simulation models



Static simulation – Monte Carlo simulation: represents a system at a particular point in time.

Dynamic simulation: represents systems as they change over time.

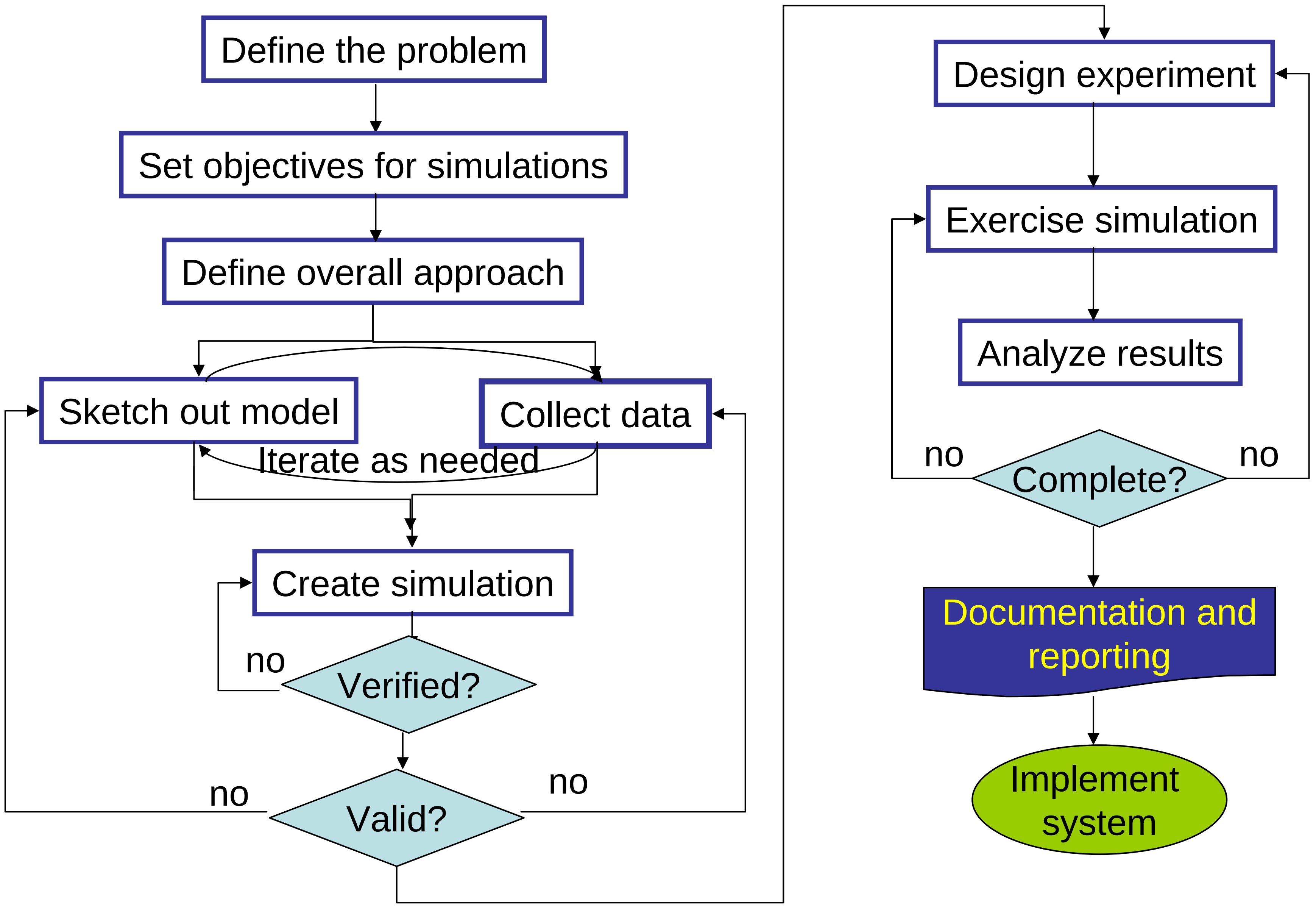
Deterministic simulation: contains no random variables. Known set of inputs will result in a unique set of outputs

Stochastic simulation: random inputs -> random outputs
outputs – estimates of the true characteristics of the model.

Couple of examples of simulation systems

- Banking system: waiting time performance for customers.
  - State variables: number of busy tellers, number of waiting customers (in the line)
  - What kind of model? Ans: discrete, stochastic, dynamic
- Production: factory performance
  - State variables: status of machines: busy, idle, down
  - What kind of model? Ans: discrete, stochastic, dynamic
- Memory-less coin flipping experiment
  - State variable: head or tail
  - What kind of model? Ans: discrete, stochastic, static
- Noise in an electronic circuit
  - State variable: current noise level
  - What kind of model? Ans: continuous, stochastic, static
- Steam engine simulation:
  - State variable: pressure
  - What kind of model? Ans: continuous, deterministic, dynamic

Steps in a simulation study



Homework no 1 – due next week in class

- Install Visual C++ 6.0 and OMNET++ and compile the token ring example, token. Read the readme file for a description of the token ring protocol. According to today's lecture define the system model, specifying all its components (entities, attributes, etc). What type of simulation model is this (out of the 8 possible described categories), and why? Is simulation appropriate for this system (Hint: relate to the criteria discussed in the class)? Provide a list of events for one particular run of the game.