# Simulation Software

## Simulation Software

- Basically three levels of software to support simulation:
  - High-level programming languages
    - Fortran, C/C++, …
  - General-purpose simulation software
    - SIMAN, GPSS, MODSIM, …
  - Special-purpose simulation packages
    - ProModel, Witness, AutoMad, OPNET …

## High-level Languages for Simulation

- Follow simulation principles discussed in chapter 3
  - For event-scheduling simulation, Figure 4.1:
    - Determine and keep track of system state, entities and attributes, sets, events, activities, and delays.
    - Keep track of system clock.
    - Define and design event routines.
    - Develop random-variate generators to generate samples from desired probability distributions.
    - Report stats.
- Widely available
- Flexible
- Maybe time-consuming

## General-Purpose Simulation Languages

- May support various applications
- May consist of multiple layers
- Contain simulation components
- Example: SLX is a layered modeling system in which GPSS/H comprises only one of the layers. These layers include:
  - Level 4: special packages for nonsimulationists, typically GUI based.
  - Level 3: application-specific packages, including manufacturing, telecommunications.
  - Level 2: next generation GPSS/H.
  - Level 1: simulation and statistical primitives and routines.
  - Level 0: kernel (largely C based) and other primitives required for simulation.
- Flexible and tailorable
- Less efficient

## Special-Purpose Simulation Software

- Designed for specific applications, such as manufacturing and health care systems.
- Also called simulator, i.e., flight simulator.
- May need special hardware support.
- Efficient
- Application specific

## Selection of a Simulation Software

- Factors to consider:
  - Training and experience of the simulation analyst
  - Nature of the application
  - Value of the simulation project relative to the purchase cost of the hardware and software