Models of Computing

In this course we will study a number of different models that appear in distributed systems:

- object model
- tuple space model
- agent model
- metasystem model
- transaction model
**Object Model**

- Encapsulated object
- Interacting objects
- Distributed objects

**Tuple Space Model**

- Processes
- Write/add
- Generate
- Read/remove
Agent Model

machine/node

local resource

agent

migrate

Metasystem model

Cluster:

very high speed, low latency network (e.g. myrinet)

Grid:
Examples of Models

- **Fixed computation-transportable data**
  - Object-based
    - CORBA (OMG)
    - RMI (Java)
    - Com/Dcom (MS)

- **Transportable computation-transportable data**
  - Metasystems
  - Globus
  - Legion
  - Agents
    - Voyager
    - Aglets
Major Topics

- Concurrency, Synchronization, Coordination
  - programming languages: Java
  - modeling/analysis languages: CCS
- Distributed Scheduling
- Fault Tolerance/Recovery
- Protection/Security

Concurrency, Synchronization, Coordination

- safety vs. liveness
  - safety: insuring consistency of system
  - liveness: insuring progress of systems activity
- conservative vs. optimistic:
  - conservative: refuse to perform any action unless the system's consistency can be guaranteed
  - optimistic: perform actions with the expectation of their successful completion and be prepared to recovery to a consistent state if they cannot be completed
Forms of Synchronization

- mutual exclusion - preventing concurrent access to shared objects to preserve the consistency of the object
- condition synchronization - blocking attempted operations on a shared object until that object is in a state where the operation will preserve the consistency of the object

Java as a Concurrency Programming Language

- Language:
  - language concepts for threads and synchronization
  - platform independent
- Libraries for basic network programming
  - sockets
  - Remote Method Invocation
- Used to implement distributed systems
  - Aglets
  - Voyager