IE630: Simulation Modelling & Analysis
Agent-based Modelling & Simulation

Saurabh Jain
Assistant Professor
Department of Industrial Engineering and Operations Research
IIT Bombay

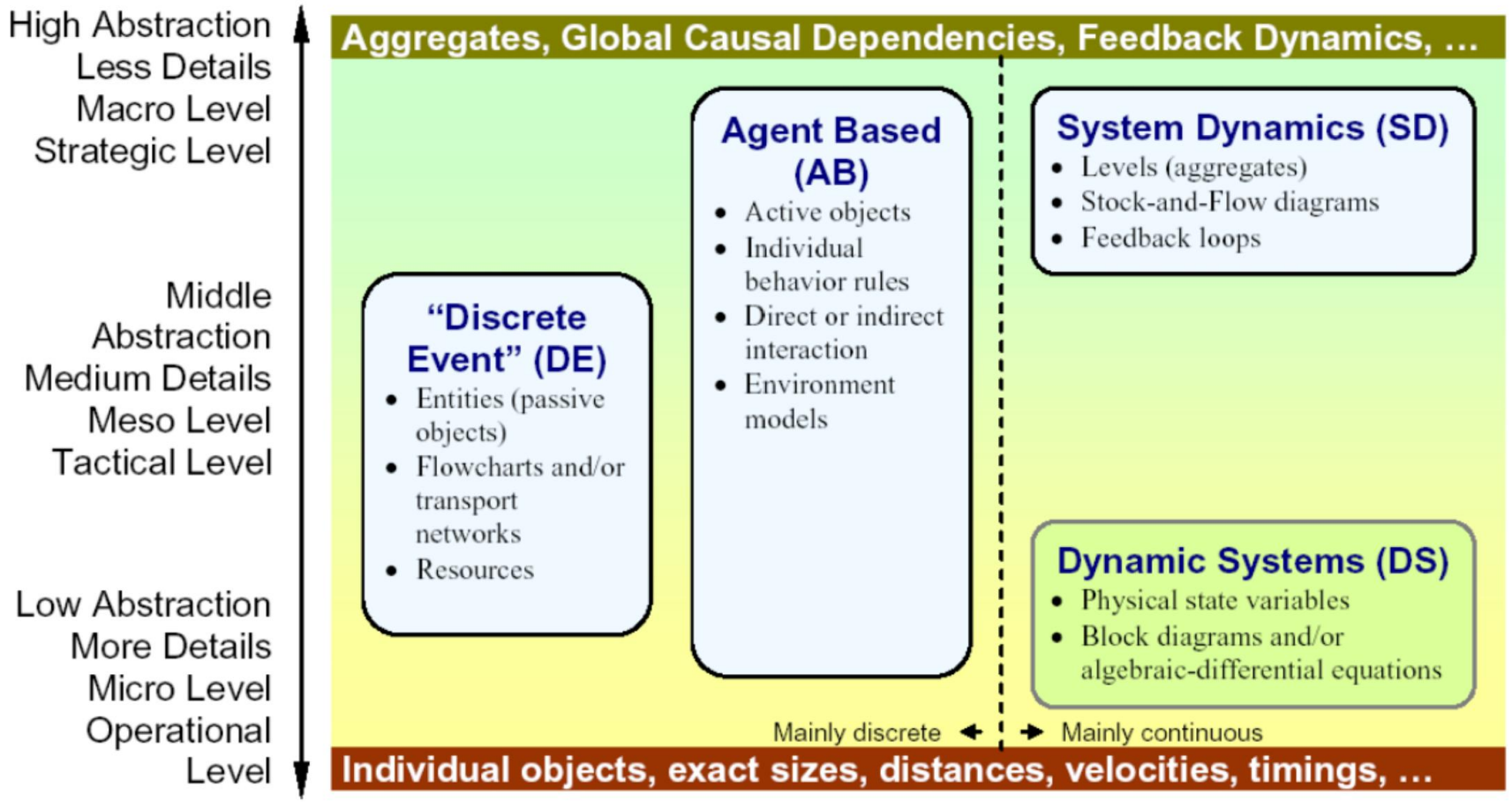
IIT BOMBAY

IEOR IIT BOMBAY

Quick Recap

Introduction

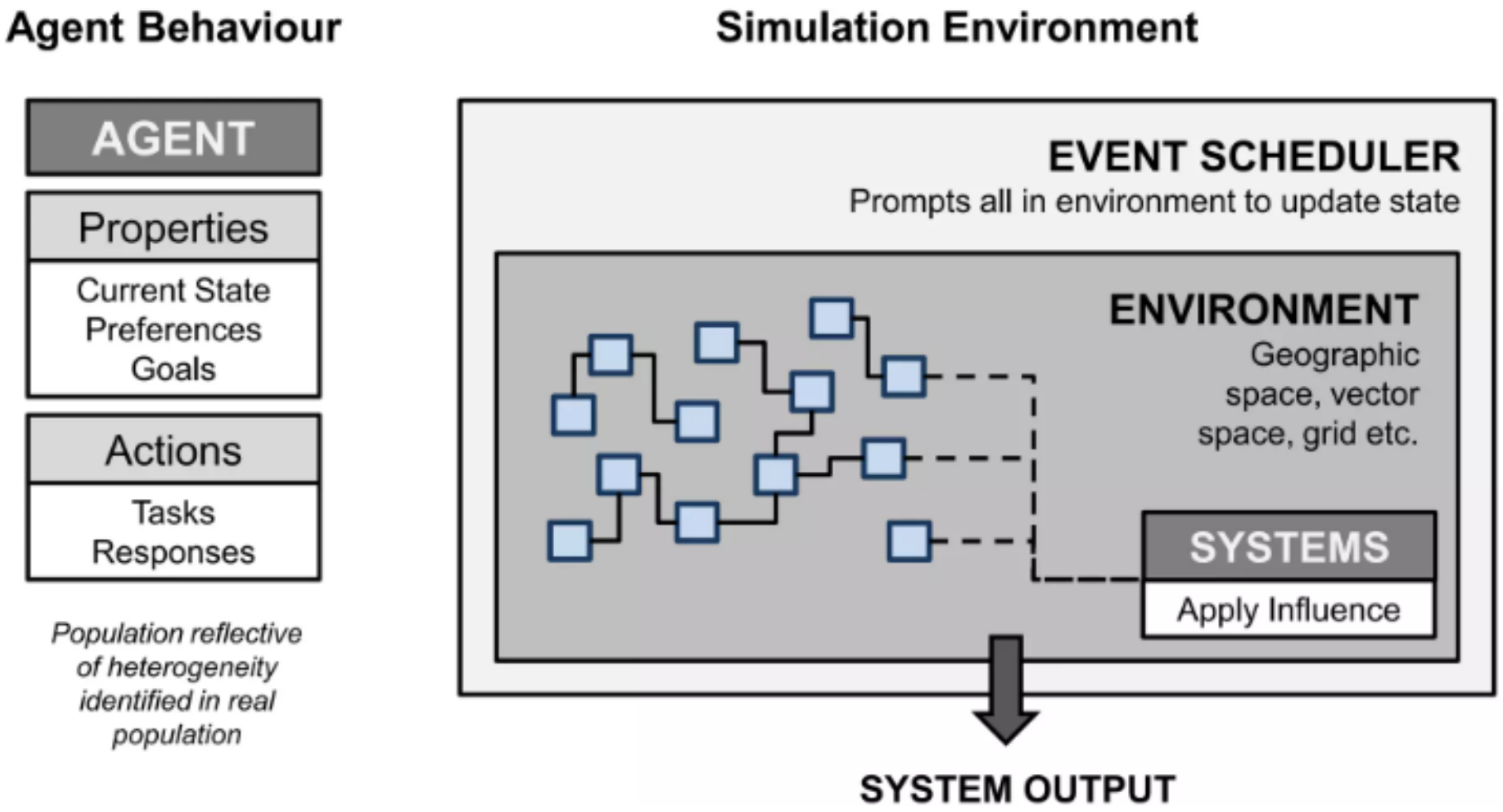
- Four paradigms of simulation
  - Discrete Event
  - Agent-based
  - System Dynamics
  - Dynamical Systems
- Abstraction Levels?



Agent-based Modelling

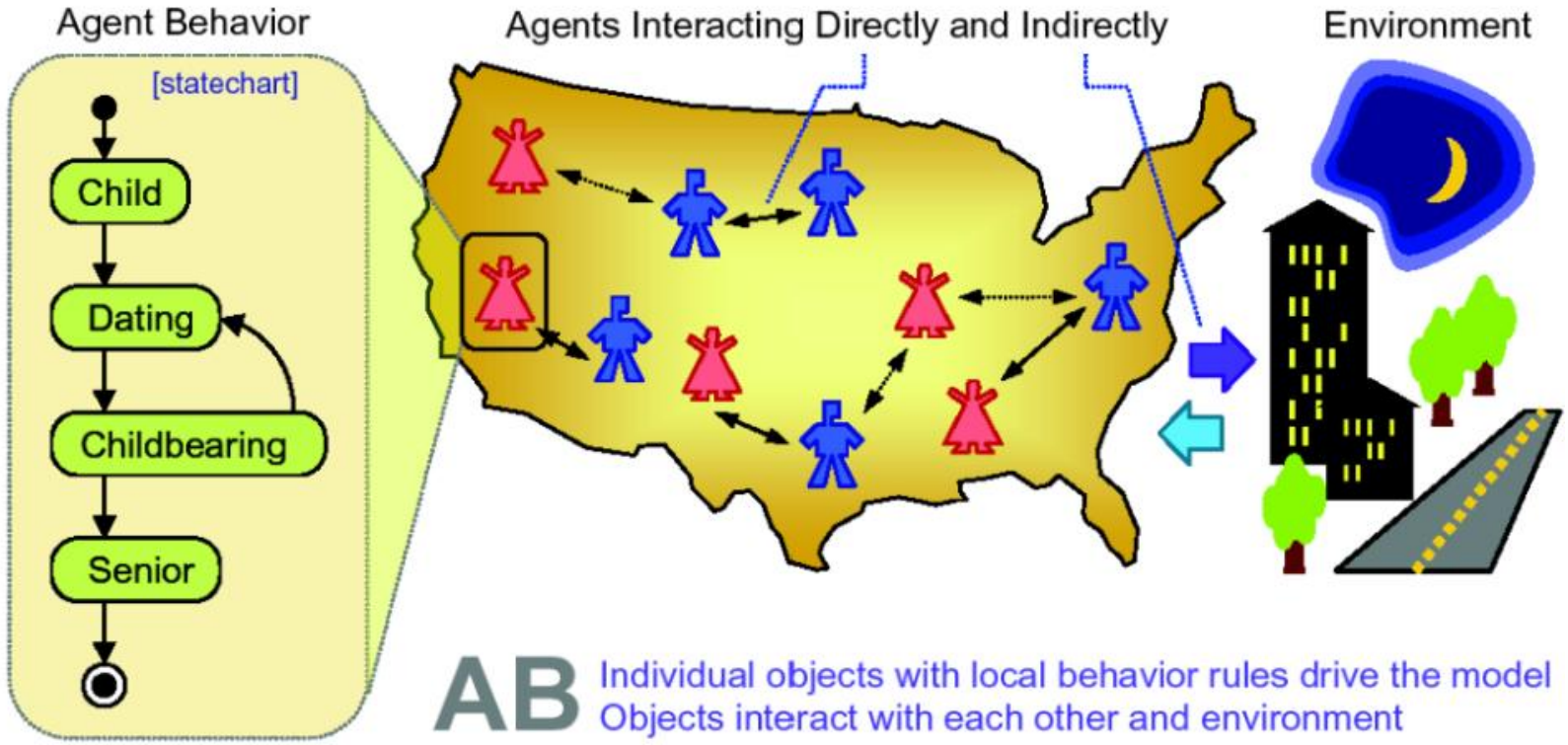
- Properties of the objects
  - Pro-activeness, ability to learn, mobility, cooperation etc.
  - Decentralized decision making
- Bottom up Modelling
- There is no place where global behaviour of the system is defined
  - ( ) => Global Behaviour
- Object oriented based : class, instances, object encapsulation
- **Components**:
  - A collection of agents and their states
  - Rules governing the interactions of the agents
  - Environment within which they live

Agent-based Modelling



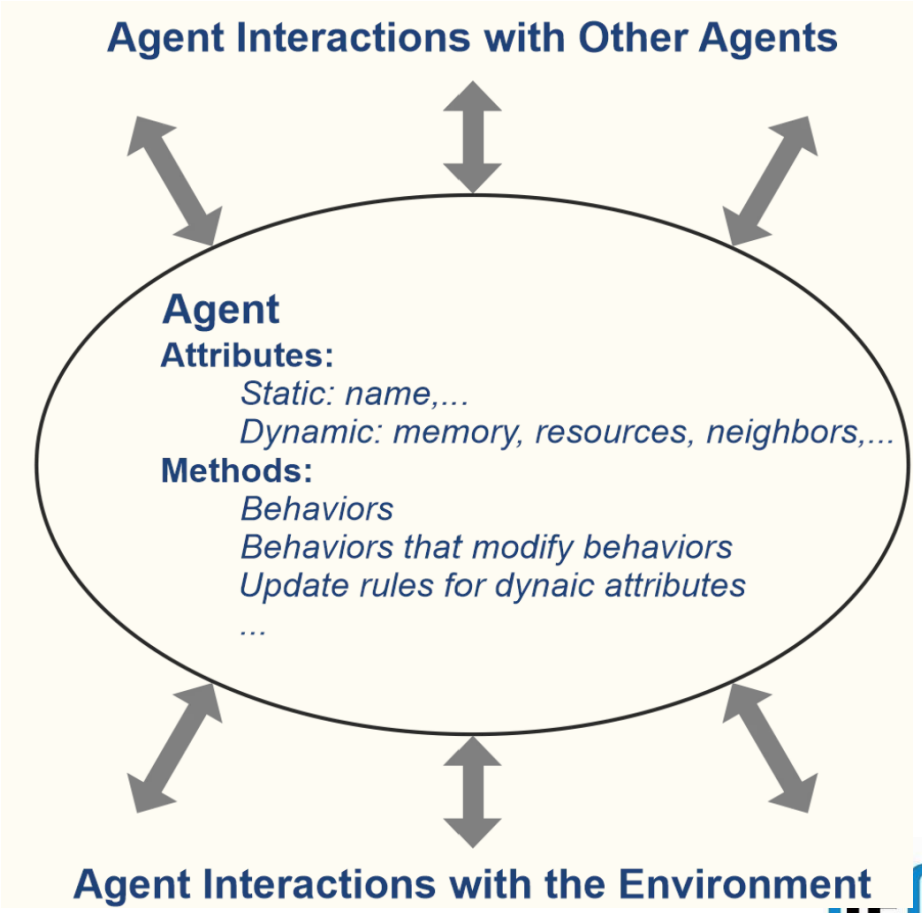
Agent-based Modelling

- **Agent-behavior:** UML state chart
- **Environment:** Housing, jobs, transport infrastructure
- **Agent:** Direct interaction with agent; indirect interaction with agent via environment



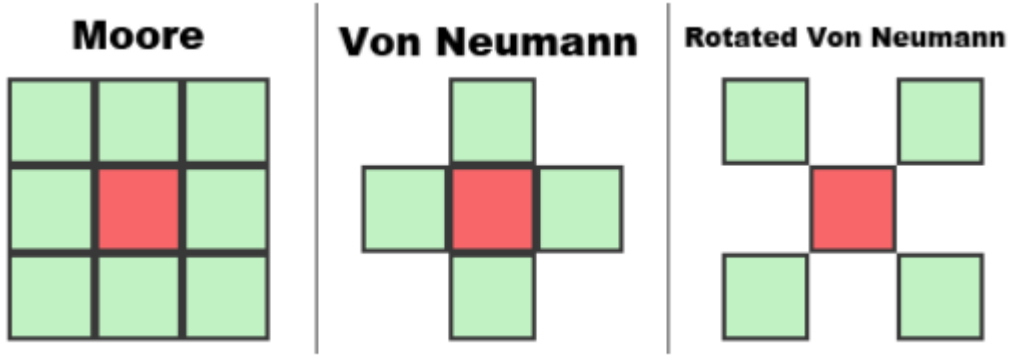
What is an Agent?

- There is no universal agreement on the precise definition of the term “agent” in the context of ABMS.
- **Properties:**
  - Encapsulated
    - Clearly identifiable, with well-defined boundaries and interfaces
  - Situated in a particular environment
    - Receives input through sensors and acts through effectors
  - Capable of flexible action
    - Responds to changes and acts in anticipation
  - Autonomous
    - Has control both over its internal state and over own behavior, reacts to environmental change and proactively changes its behavior
  - Designed to meet objectives
    - Attempts to fulfill a purpose, solve a problem, or achieve goals

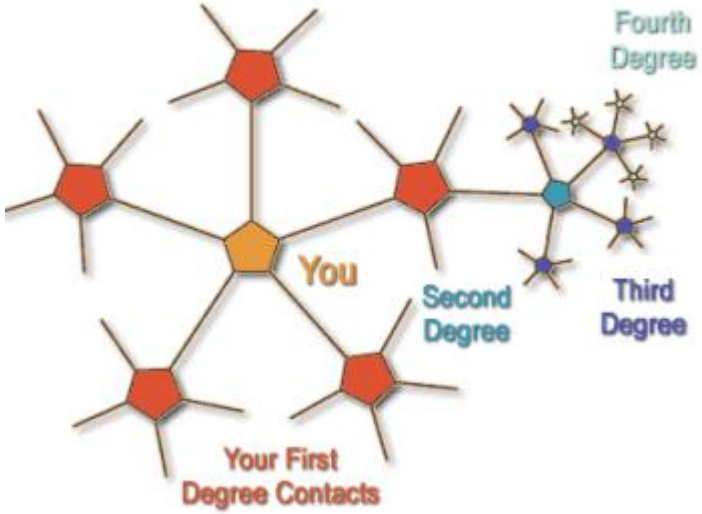


Agent Interactions

- An underlying topology of connectedness defines how and with whom the agent interacts

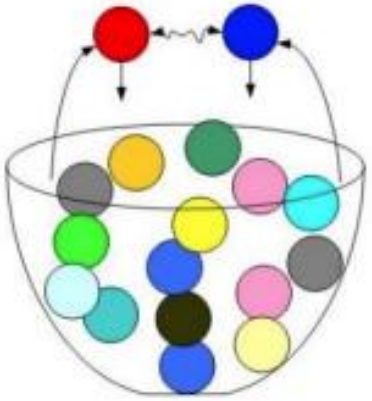


- Depending on the environment, agent interacts with a subset of other agents.

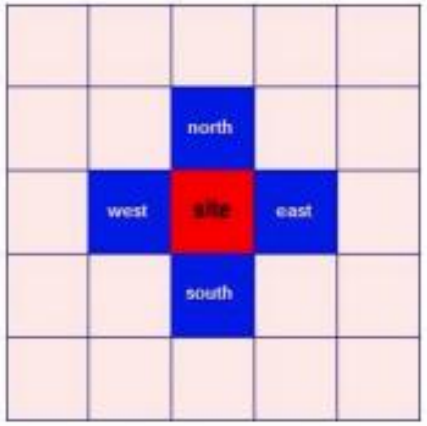


- Typically, an agent interacts with its neighbors

Agent Environment

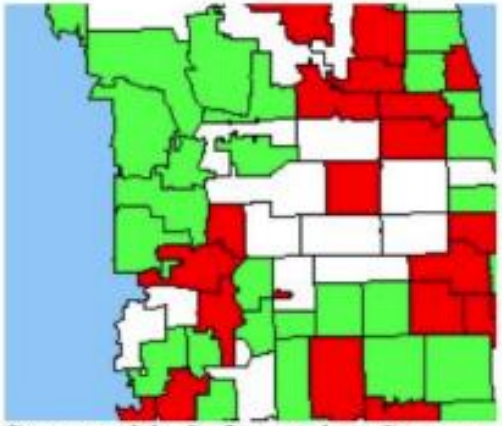


(a) “Soup” Model (Aspatial)

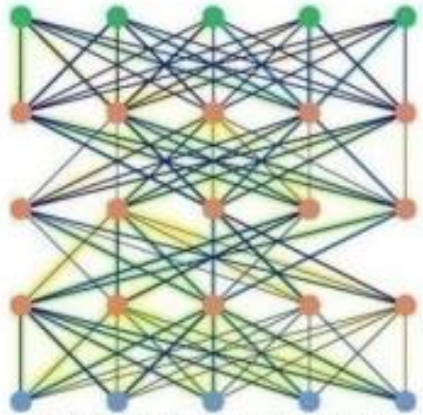


(b) Cellular Automata (von Neumann)

(c) Euclidean Space (2-D)



(d) Geographic Information System (GIS)



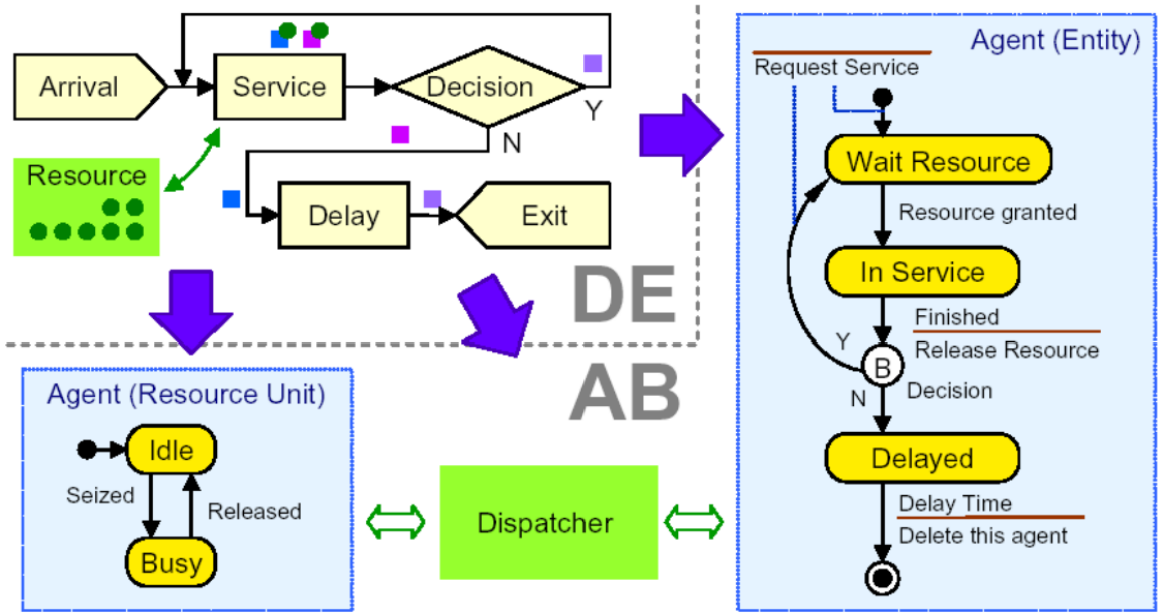
(e) Network topology

Correspondence between DE and AB models

- It only makes sense to do this in case we wish to model some extra individual behaviour later on that is hard to capture in DE style

- **Alternative 1:** Use of dispatcher (environment model): indirect communication between agents, e.g. queueing

- **Alternative 2:** agents may see each other and communicate directly to manage resource access

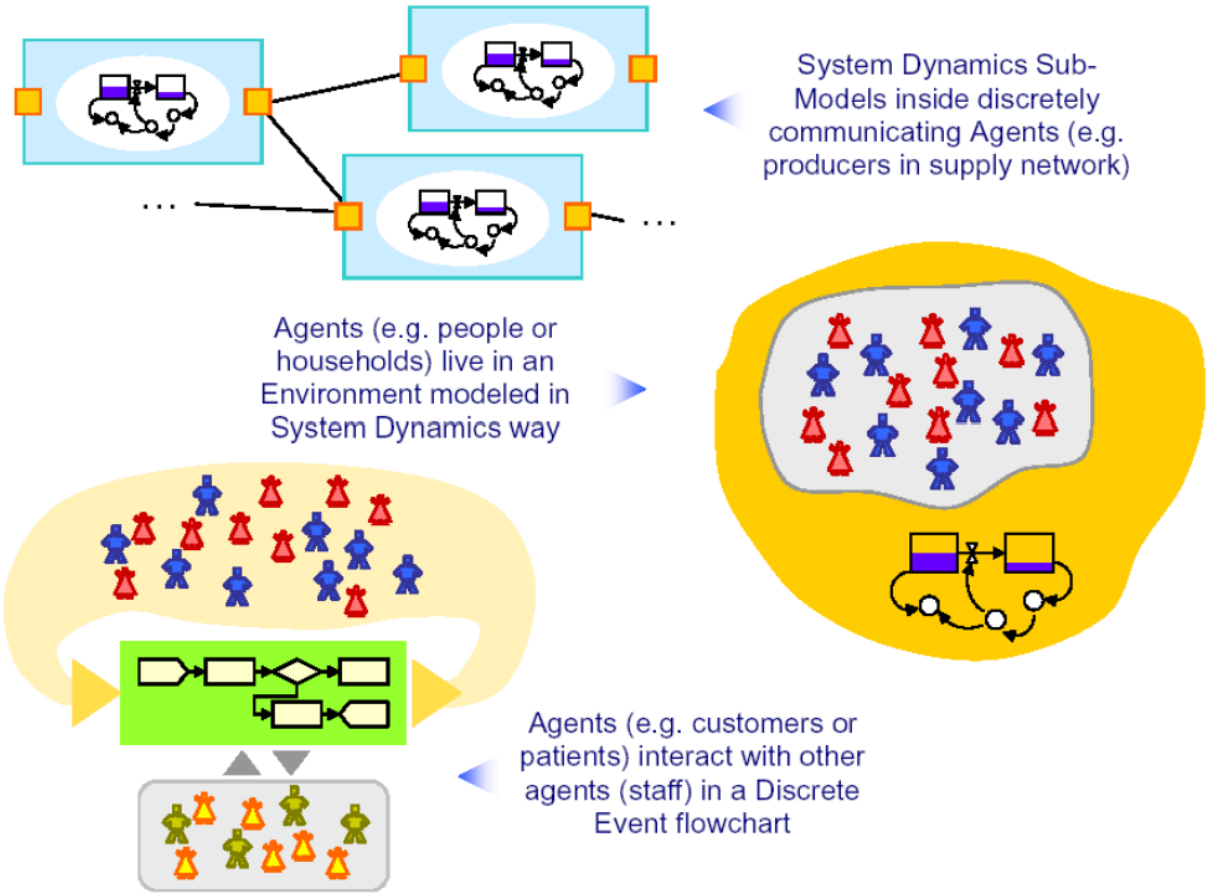


Look at the process from an entity (or resource unit) viewpoint.
Each entity (resource unit) becomes an agent.
A kind of Dispatcher may be needed to arrange interactions.

Multi-Paradigm Hybrid Modelling & Simulation

- Supply Chain
  - Intra supply member process are SD
  - Communication between members are AB

- Hospital
  - Agents interact with other agents in a DE flowchart
  - Like Dispatcher: Overall interaction is somewhat “centralized”;



Software

Platform	Scalability	Execution Speed	Programming Language	Primary Domain	Web site
NetLogo	desktop computing	intermediate	NetLogo	social and natural sciences	www.ccl.northwestern.edu/netlogo/
MASON	large-scale	fast	Java	social complexity, physical modeling, AI/machine learning	www.cs.gmu.edu/~eclab/projects/mason/
Swarm	large-scale	slow	Objective-C; Java	general purpose	http://alumni.media.mit.edu/~nelson/research/swarm/
Repast	large-scale	fast	Java; Python; C++	social sciences	http://repast.sourceforge.net/
Ascape	large-scale	fast	Java	general purpose	http://ascape.sourceforge.net
AnyLogic	large-scale	fast	Java	general purpose, distributed simulation	www.anylogic.com