CmpE 593
Multiagent Systems

Pınar Yolum
pinar.yolum@boun.edu.tr

Department of
Computer Engineering
Boğaziçi University

Multiagent Systems

Based largely on

Service-Oriented Computing: Semantics, Processes, Agents
– Munindar P. Singh and Michael N. Huhns, Wiley, 2004

Agent Abstractions/1

- The traditional abstractions are from AI and are mentalistic
  - *beliefs*: agent's representation of the world
  - *knowledge*: (usually) true beliefs
  - *desires*: preferred states of the world
  - *goals*: consistent desires
  - *intentions*: goals adopted for action

Agent Abstractions/2

- The agent-specific abstractions are inherently interactional
  - *social*: about collections of agents
  - *organizational*: about teams and groups
  - *ethical*: about right and wrong actions
  - *legal*: about contracts and compliance

Agent Abstractions/3

Agents, when properly understood

- lead naturally to multiagent systems
- provide a means to capture the fundamental abstractions that apply in all major applications and which are otherwise ignored by system builders

Agents versus AI

	Traditional AI	Agents
Entities	Stand-alone	Social: flexible autonomy, communities, responsibility
Actions (in terms of)	Cause and effect	Ethical concepts of right and wrong
Contracts (in terms of)	Simplistic obligations	Directed relationships capturing rights, duties, powers, and liabilities.

How to Apply the Abstractions

Consider how the components of a large and dynamic software system in a practical situation

- Dynamism => autonomy
- Openness and compliance => ability to enter into and obey contracts
- Trustworthiness => ethical behavior

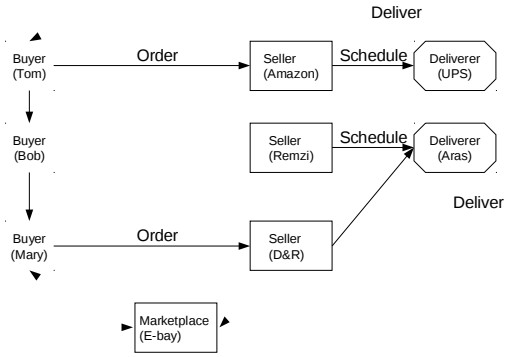
Why Do These Abstractions Matter?

- Because of modern applications that demand going beyond traditional metaphors and models
  - *Virtual enterprises*: manufacturing supply chains, autonomous logistics,
  - *Electronic commerce*: utility management
  - *Communityware*: social user interfaces
  - *Problem-solving* by teams

Multiagent Applications (1)

- Web Intelligence
  - Personal assistant that searches books for you, picks a good one for your needs, and pays with your credit card
  - An agent-based peer in the P2P networks that models your taste in music, gets recommendations, connects to other peers, and downloads MP3s
  - Trust as an abstraction

MAS Example



Multiagent Applications (2)

- Ambient Intelligence
  - Agent in your refrigerator keeps track of your groceries (ex. LG Internet refrigerator).
  - Agent in your cell-phone keeps track of your location.
  - When you are passing in front of a Migros, your agent in the cell phone asks your refrigerator agent if you need groceries
  - Cell phone agent alerts you to stop at the market

Multiagent Applications (3)

- Business Intelligence
  - Agent carries out the transactions for a company
  - Negotiates with the customer for B2C transactions and with supporting service providers for B2B transactions
  - Negotiation in money, value, goods

Multiagent Applications (4)

- Business Intelligence
  - A customer agent participates in an auction for its user
  - Bids for individual goods or combination of goods in auctions
  - Sells some of its goods to other agents
- Trading Agent Competition

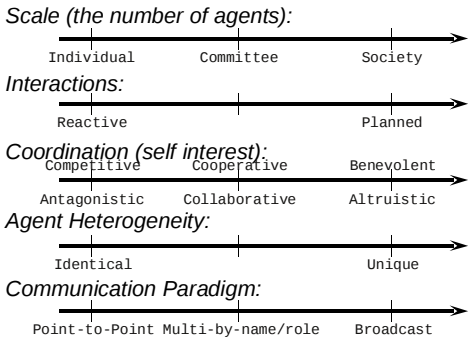
Multiagent Applications (5)

- Intelligent Healthcare
  - Agent runs on the Alzheimer patient's handheld.
  - Using a GPS, the agent keeps track of and learns where the patient goes in her everyday routine.
  - If she starts going places out of the ordinary, the agent locates the agent of a relative
  - Communicates the situation to the agent
- Usability concerns

Attributes of Multiagent Systems

- Decentralization
  - No central control
- Complex components
  - Each agent autonomously decides on actions
- Adaptive behavior
  - Agents learn certain behavior over time
- Complex interactions
  - Not rigid like typical distributed systems
- Coordination
- Emergent, aggregate behaviors
  - The interactions bring out interesting properties about structure or behavior

Dimensions of MAS: System



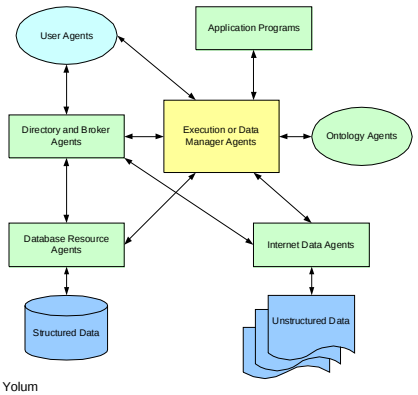
Challenges of MAS (1)

- Organization
  - Distribution of control among agents
  - Finding others (e.g., through directories)
  - Consistency maintenance, and reconciliation of conflicts among agents
  - Enforcing organizational rules
- Representation
  - Tasks, goals, etc.
  - Environment
  - Other agents
- Coordination
  - Description,
  - Decomposition,
  - Distribution of tasks among agents
  - Application: Business process modeling and enactment

Challanges of MAS (2)

- Communication
  - Agent communication languages
  - Interaction protocols
  - Semantics and pragmatics
- Service Selection
  - Whom to *trust*
  - Modeling others
  - Protocols for
- Methodologies for developing large systems
  - Abstractions

Standard Agent Types: Realizing a MAS



Name Service

- A multiagent architecture requires scalable, symbolic name resolution
- Alternative naming protocols
  - FIPA
  - LDAP
  - Jini
  - CORBA Naming Service
  - JNDI

Directory Service

- Simple yellow-page service
- Registered agents advertise their services by providing their name, address, and service description
- Agents request recommendations for available services (provided by other registered agents or services)
- A simple database-like mechanism that allows agents to
  - insert descriptions of the services they offer
  - query for services offered by other agents
- Brokerage, recruitment and mediation services are not provided by Directory Service
- UDDI for Web services

Brokerage Service

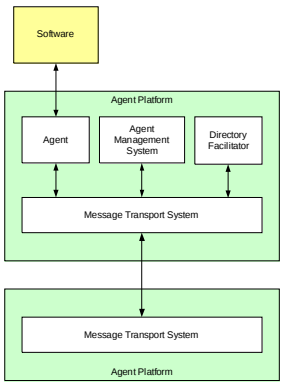
- Cooperates with a Directory Service
- An agent requests the Brokerage Service to recruit one or more agents who can provide a service
- Brokerage Service uses knowledge about the requirements and capabilities of registered agents to
  - Determine the appropriate agents to which to forward a request for a service
  - Negotiates with the agents to determine a suitable set of service providers
  - Potentially learn about the properties of the responses
    - example: Brokerage agent determines that advertised results from agent X are incomplete and seeks a substitute for X

FIPA-Compliant Agent Frameworks

- FIPA is the Foundation for Intelligent Physical Agents, with website at www.fipa.org
- Specifies standards for heterogeneous, interoperating agent-based systems.

- Some of the popular, FIPA-compliant agent frameworks used for designing multiagent systems:
  - FIPA-OS, http://fipa-os.sourceforge.net/
  - JADE, http://sharon.cselt.it/projects/jade/
  - Zeus, http://zeus.enhydra.org/

FIPA Agent Management Reference Model



FIPA Agent Management Services

- Agent Management System
  - Mandatory component
  - A single instance exists in the system
  - Authority among all existing agents
  - Agents register with AMS
  - Functionalities: register, deregister, modify, search, get-description
- Message Transport Service

FIPA Agent Management Services

- Directory Facilitator
  - Optional component
  - Yellow-pages service
  - Dedicated, trusted facilitator that keeps an accurate list of agents in the system
  - Many DFs can exist in the same system
  - Agents register and deregister from any number of facilitators in the system
  - Agents can modify their listing in a facilitator
  - Agents can search the directory
    - First locally, then to other DFs in the system
    - Using DFS

Constructing Agent-Based Systems

- Development environments provide FIPA services and agent skeletons
- Examples: Jade, Zeus
- Jade
  - Implements FIPA Agent Management Services
  - Messages are FIPA ACL messages
  - Each platform has one JVM; agents are threads
  - Each agent is controlled by a scheduler for selecting, executing, managing behaviors
  - Scheduler executes non-preemptive round-robin