Multiagent Systems for Service-Oriented Computing

- Challenge: Organizing a decentralized computation
  - What services constitute a service engagement
  - Who provides what services to whom
  - Without the benefit of a central designer for all services
- Solution: Interacting and communicating
  - Trade off prior agreement with formal reasoning about specifications
  - Specify interaction protocols that describe desired interoperation
  - Design agents to participate in specified protocol
  - Potentially enable agents to negotiate agreements dynamically
- Specialized protocols
  - Negotiation
  - In cooperative, homogeneous setting: maintaining consistency

Agents in Service-Oriented Computing

Breakdown of functionality

- User assistance
- Application adapters
- Directory and ontology
- Brokerage
- Resources: Web, databases, . . .
- Process planning and execution

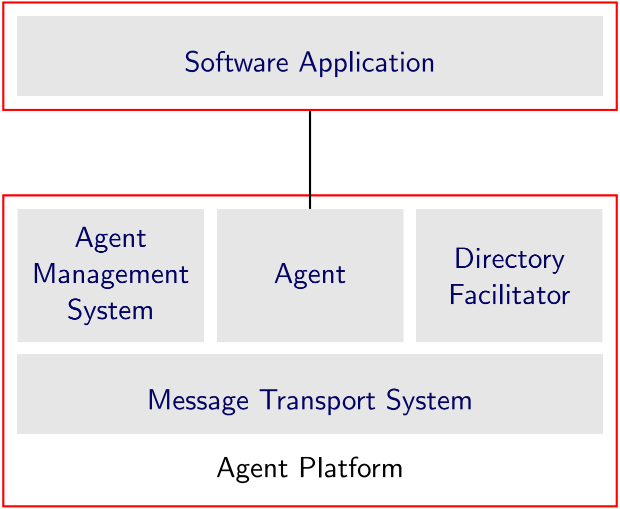
Brokerage

- Cooperates with a Directory Service
- Accepts requests from agents to recruit one or more agents who can provide a service
- Uses knowledge about the requirements and capabilities of registered agents to
  - Identify appropriate agents for an interaction
  - Negotiate with selected agents
  - Potentially learn models of the responses
  - Example: Brokerage determines that advertised results from agent X are incomplete and seeks a substitute for X

FIPA Agent Management System

Foundation for Intelligent and Physical Agents (now in IEEE)

- Good: architecture
  - Highlights agents and interaction
- Wrong: mentalist focus
- Wrong: Over-constrained protocols
- Wrong: Already obsolete low-level details



Agent Management System Functions

Analogous to a Java Enterprise Edition Container

Handles the creation, registration, location, communication, migration, and retirement of agents

- White pages, e.g., agent location and naming
    - Agent identifiers support social names, transport addresses, name resolution services
- Yellow pages, e.g., service location and registration services, from Directory Facilitator
- Agent message transport services

Multiagent Frameworks

- JADE, a popular FIPA-compliant agent framework for multiagent systems:
  - http://jade.tilab.com/
- Jadex: JADE plus BDI constructs
- JaCaMo: Combines three programming approaches
  - Jason: BDI constructs
  - Cartago: Environment artifacts
  - Moise: Organizations (later Moise+)
- Janus http://www.janusproject.io/
  - Comes with the SARL agent-oriented programming language
- Inactive projects: FIPA-OS, Jack, Zeus

Summary: Multiagent Systems

Interactions among agents enable interoperation necessary in service engagements

- Communication among agents is key
- Programming environments can support agent interactions
- In cooperative settings, consistency maintenance is a useful utility
- To intelligently cooperate or compete, agents must model each other
    - Such modeling requires complex representations and reasoning
- The guarantees we achieve without relying upon agent internals are the most robust
    - Correspond to interaction protocols for interoperation
    - Yield loose coupling
    - . . . The next topic