

CmpE 593 Multiagent Systems

Pınar Yolum
pyolum@cmpe.boun.edu.tr

Department of
Computer Engineering
Boğaziçi University

Chapter 8 Engineering

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

Characteristics of Agent-Based Systems

- Roles
 - Agents enact roles that define the necessary capabilities and commitments
- Capabilities
 - Functionalities that the roles can support
- Reconfigurability
 - Agents' location is transparent to the rest of the system
 - Found through lookups on the naming registries

Characteristics of Agent-Based Systems

- Caution
 - No guarantees about others in the system
 - Usual security considerations
 - Trust considerations
- Abstraction
 - Expose little about the internals of the agent
 - Internals can be changed easily
- Loose coupling
 - Avoid unnecessary dependencies between agents
 - Computations done autonomously

Design of Agent-Based Systems

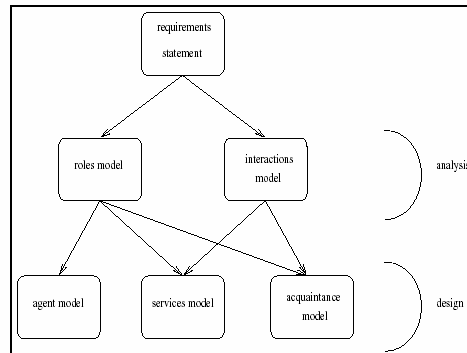
- Agent-Oriented Software Engineering (AOSE)
- Incorporate characteristics of agent-based systems into design
- Develop methodologies for designing agent-based systems
- Examples: GAIA (Wooldridge *et al.*, 2000), Tropos (Kolp *et al.*, 2000), AUML (Bauer *et al.*, 2001)

GAIA (Wooldridge *et al.*, 2000)

- Agents enact roles and communicate using interaction protocols
- Roles have
 - Responsibilities: What should a role do?
 - Liveness
 - Safety
 - Permissions: What can a role do?
 - Activities: Things that agents can perform
 - Protocols: Allows interactions with other agents

GAIA (Wooldridge *et al.*, 2000)

- Derive agent types
 - One or more roles for each agent type
- Specify service models
 - Inputs, outputs, preconditions, post-conditions
 - Associate services with agents
- Create acquaintance models
 - Define communications paths between agents



Spring 2005— Pinar Yolum

7

Engineering Cooperation

- Agents must coordinate their activities to carry out tasks
 - To improve their interest: Bid correctly in an auction
 - To satisfy group goals: Pass the ball to the available player in a soccer game
- AND/OR graph for representing goals
 - Representation of dependencies among goals
 - Resources to solve the primitive goals

Spring 2005— Pinar Yolum

8

Engineering Cooperation

- Define the AND/OR tree
 - Requires understanding the dependencies among goals
- Assign regions of the tree to agents
- Control decisions about which areas to explore, giving priorities to some agents when appropriate
- Traverse the graph
- Allow reporting of successful traversal

Spring 2005— Pinar Yolum

9

Task Decomposition

- By the system designer
 - Must be available during implementation
 - Static
- By the agents
 - Using hierarchical planning
- Using AND/OR tree if the task decomposition is inherent
- Spatially if the layout of information sources is significant
- Functionally if expertise distribution of agents is significant

Spring 2005— Pinar Yolum

10

Task Distribution

- **Market Mechanism**
 - Task matched to agents by generalized agreement
- **Negotiation**
 - Task assignments are negotiated among agents
- **Multiagent Planning**
 - Planning agents decide the task distribution
- **Organizational Structure**
 - Based on roles already available in the organization

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

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

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

Exceptions

- Exception: Deviation from the normal behavior
- Occurs at different levels
 - Programming: Divide by zero
 - System: Broken network connection
 - Management: Aborted transaction
 - Semantic: Precondition of a service not met
 - Pragmatic: Loss of a shipment

Exception Handling

- Before the fact
 - To eliminate exceptions
 - Use heuristics to detect things that can potentially go wrong
- After the fact
 - To detect and resolve exceptions
 - Contingency plans
- Richer representations of processes are needed to reason about exceptions

Scalability (1)

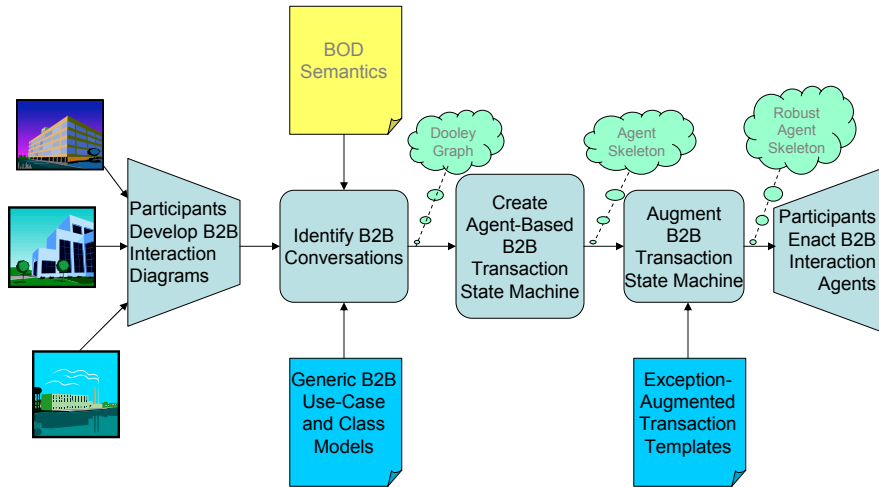
In an agent-based system

- Increasing resources for agents should increase performance
- Increasing number of agents that provide and consume services should not decrease performance
- Improving the capabilities of agents should not decrease performance

Scalability (2)

- Distribute agents across multiple machines using distributed technologies, CORBA or Java RMI
 - Introduces communication latency
- Introduce redundancy by replicating agents
 - Have multiple directory agents
 - Introduce inconsistency
- Schedule agents to execute when needed

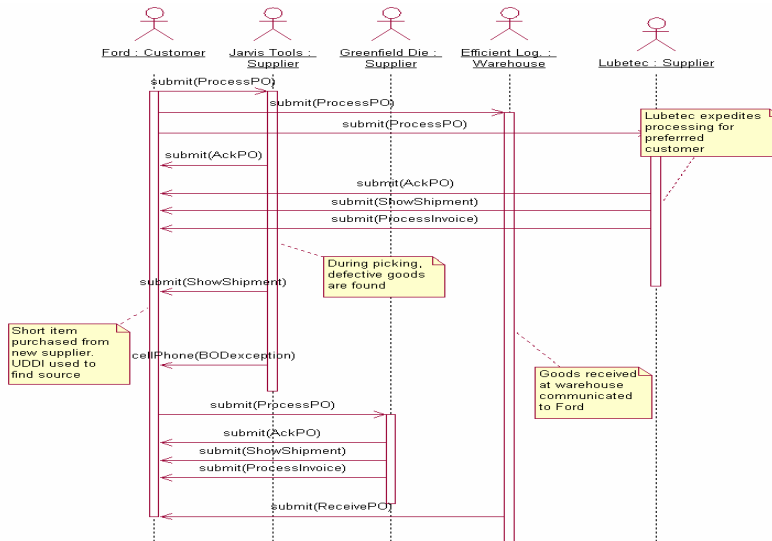
Agent-Based Coordination Methodology



Spring 2005— Pinar Yolum

19

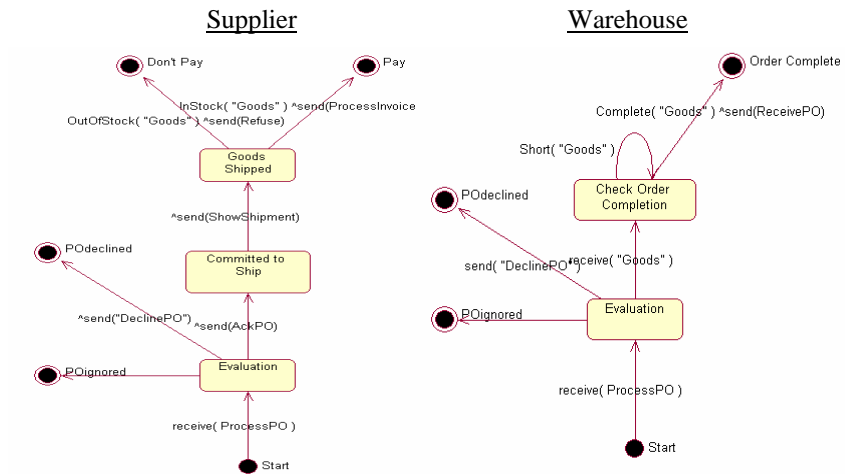
Ford Interoperability Scenario



Spring 2005— Pinar Yolum

20

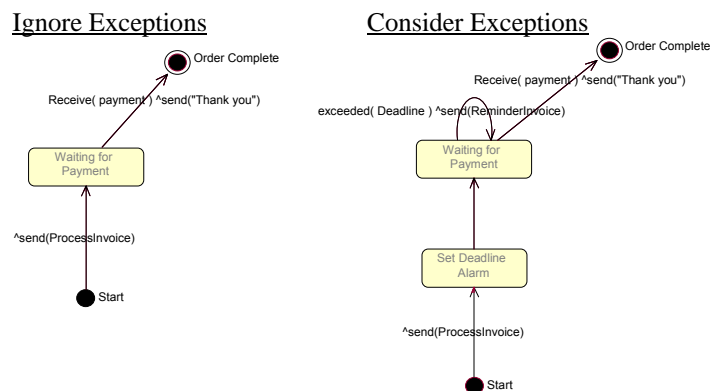
Transaction Skeletons for Agent Enactment



Spring 2005— Pinar Yolum

21

Augmented Transaction Skeletons for Agent-Based Exception Handling



Spring 2005— Pinar Yolum

22