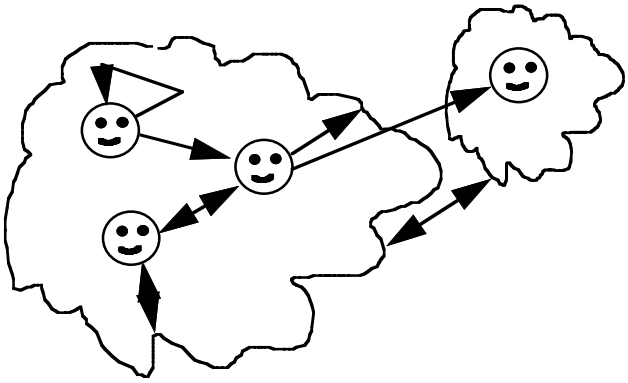
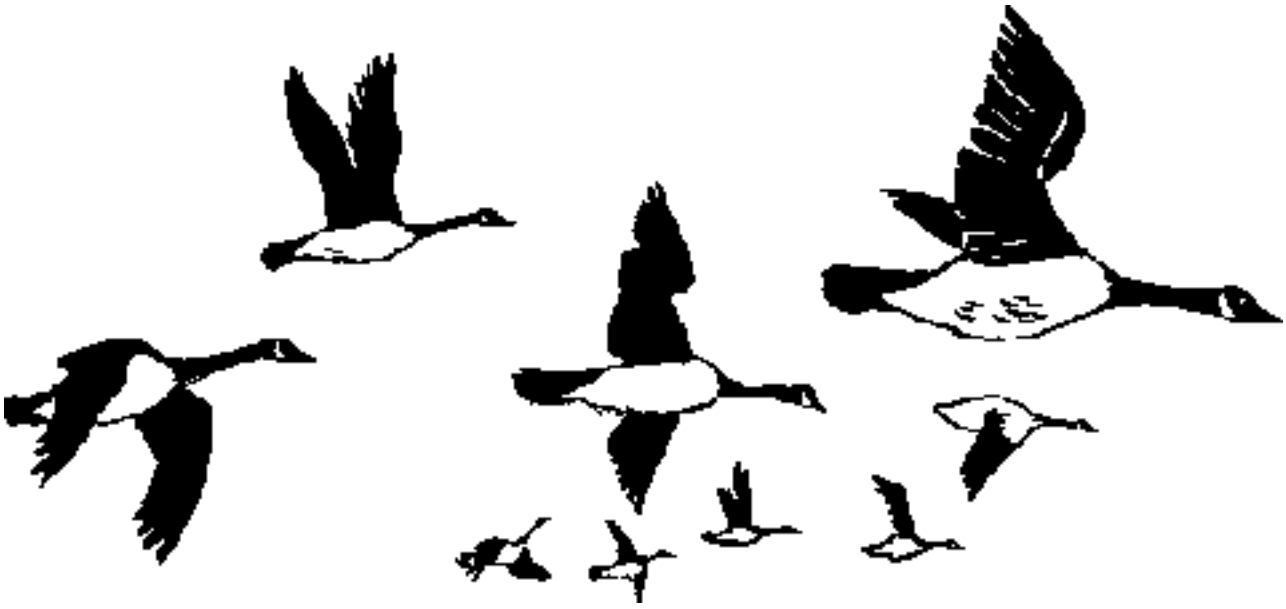
Agents



COMPLEX ADAPTIVE SYSTEMS

A way of thinking about autonomous agents and their collective behavior

It is the study of the behavior of collections of “simple” units
(e.g., atoms, molecules, neurons, people)
that have the potential to evolve over time and
give rise to coherent collective behavior.



A FLOCK IS NOT A BIRD

Most bird flocks have no leader(s). Such flocks are examples of "self organization."

Each bird basically reacts to birds nearby by following a set of simple rules.

In the "boids" simulation of Craig Reynolds, each bird had *three simple rules of behavior*:

1) Maintain a minimum distance from other objects, including other boids.

2) Try to match velocities with other boids.

3) Try to move towards the perceived center of the mass of boids in its neighborhood.

Craig Reynolds, DreamWorks SKG, Los Angeles (formerly from Symbolics Corp.); http://hmt.com/cwr/boids.html

AGENT

Dictionary Definition

Something that acts

Some common properties of agents

- **Autonomous** - capable of initiating action independent of any other entity; operates without direct intervention externally.
- **Communicative** - communicates with other agents, such as humans, machines, and software agents (i.e., social).
- **Adaptive/reactive** - sensing and acting; responds in a timely fashion to changes in the environment.
- **Mobile** - able to transport itself from one environment or platform to another.
- **Proxy**- may act on behalf of someone or something.
- **Proactive** - goal-oriented, purposeful; does not simply act in response to the environment.
- **Intelligent** - state is formalized by knowledge (i.e., beliefs, goals, plans, assumptions) and interacts with other agents using symbolic language.
- **Temporally continuous** - is a continuously running process.
- **Character** - believable "personality" and emotional state.
- **Able to learn and evolve** - learning; changes its behavior based on its previous experience.

AGENT

Basic Definition

An autonomous entity that can interact with its environment

- This is the most general sense of agent.
- Agents are not particularly useful unless they are autonomous, communicative, and reactive—and arguably mobile.

SOFTWARE AGENT

An autonomous software entity that can interact with its environment

In other words, they are agents implemented using software.

- ❏ They are autonomous.
- ❏ They can react to other entities—including humans, machines, and other software agents in various environments and platforms.

SOFTWARE TECHNOLOGY AND AGENTS

	"Monolithic" Programming	Structured Programming	Object-Oriented Programming	Agent Programming
Unit Behavior	External	Local	Local	Local
Unit State	External	External	Local	Local
Unit Invocation	External	External (CALLed)	External (message)	Local (rules, goals)

Software history is one of increasing localization and encapsulation.

POSSIBLE ANATOMY OF AN AGENT

messages via

event occurrences:

assertions

queries

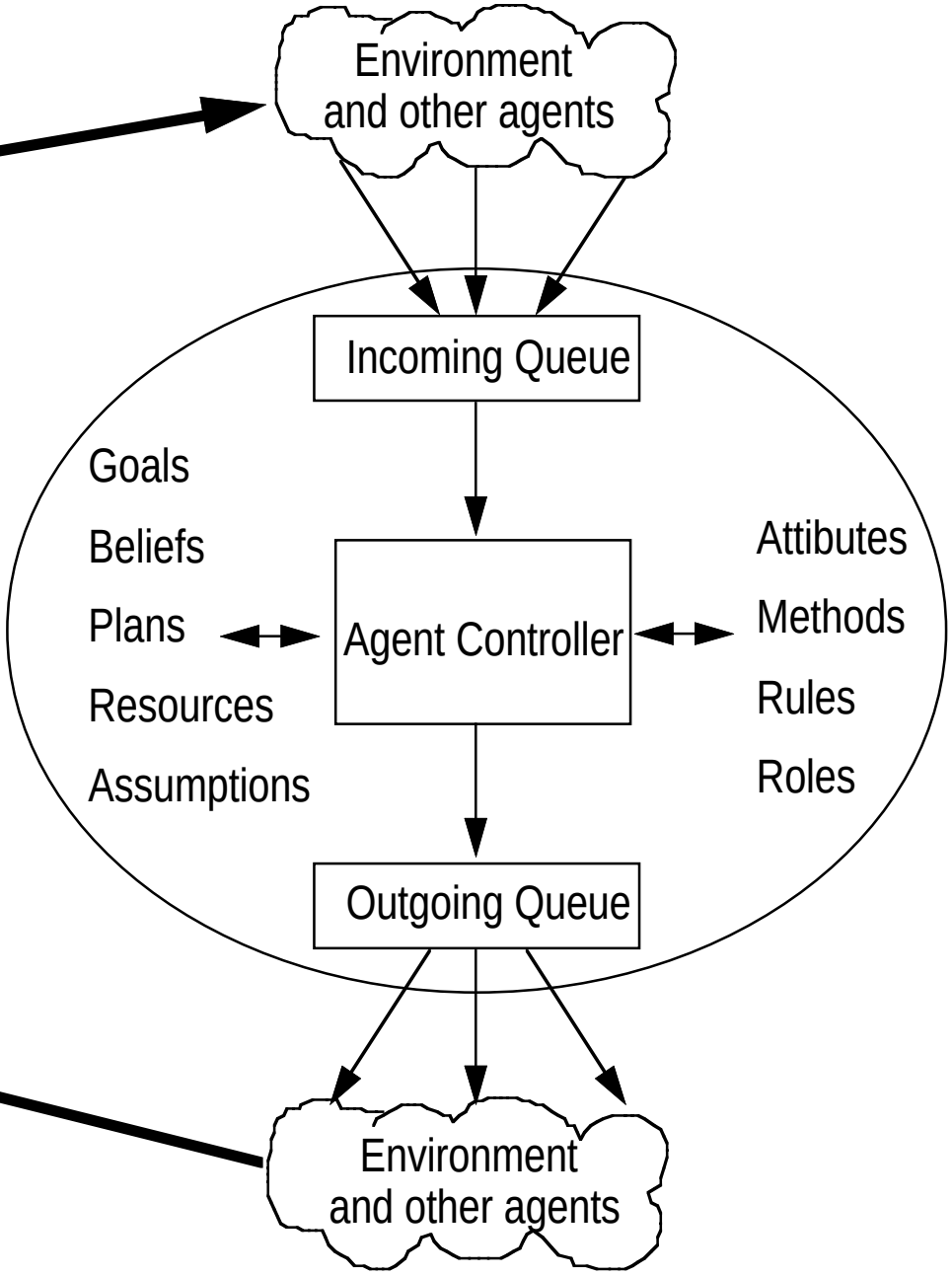
commands

state changes

other

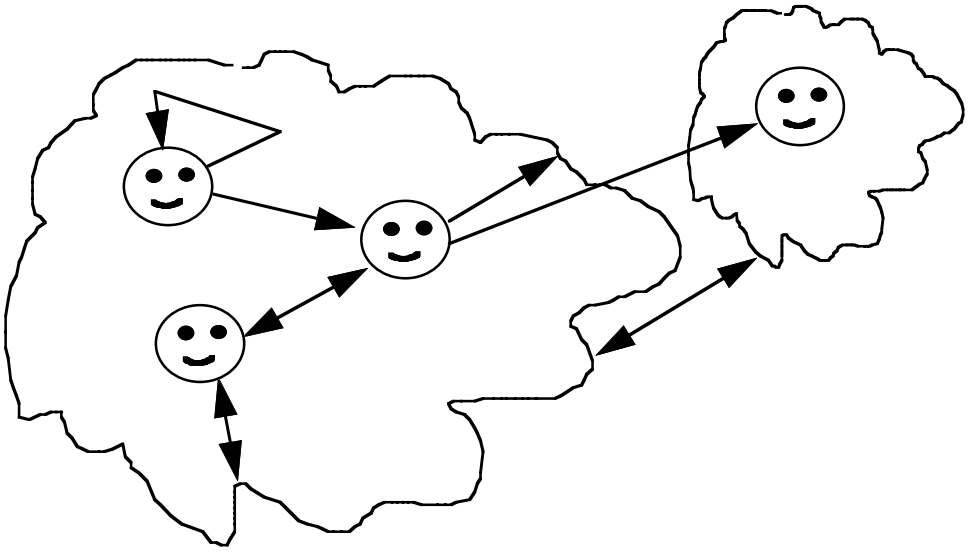
- advertise
- monitor
- register
- cancel
- subscribe
- advertise
- recruit
- broker
- tell/untell

. . .



AGENT COMMUNICATION

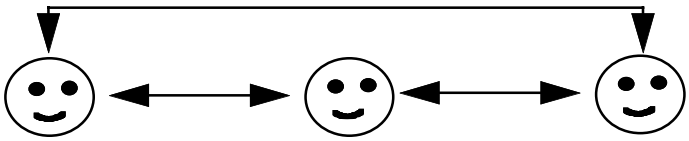
Agents can communicate with other agents—as well as their environment.



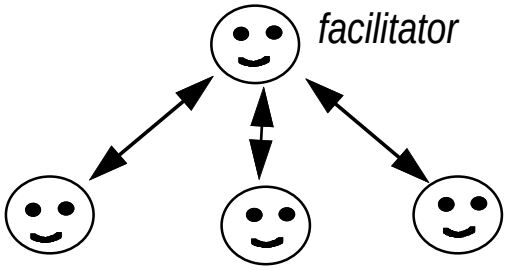
In fact, the environment itself can be treated as an agent, when appropriate.

INTERAGENT COMMUNICATION

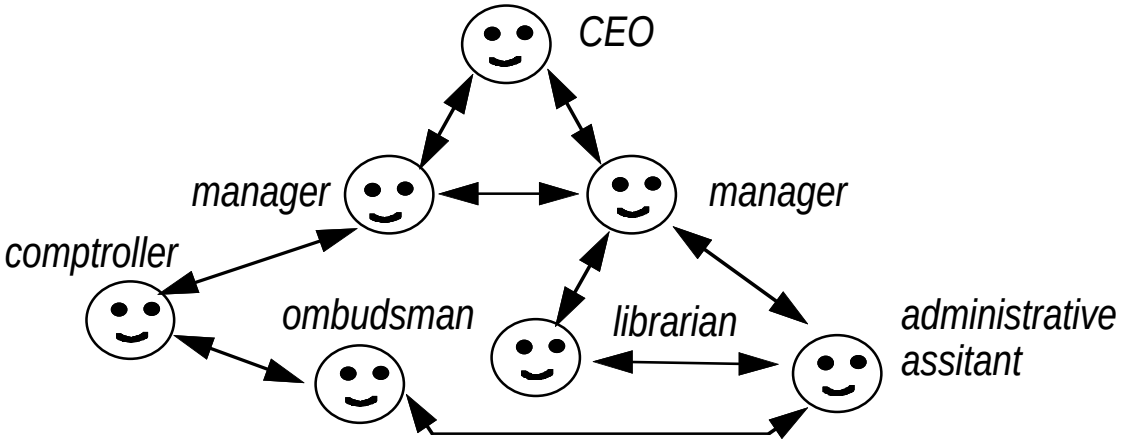
Uniform-agent architecture



Federated architecture



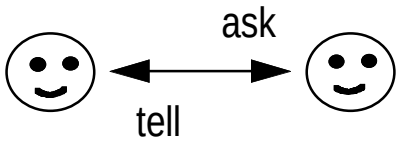
Highly specialized architecture



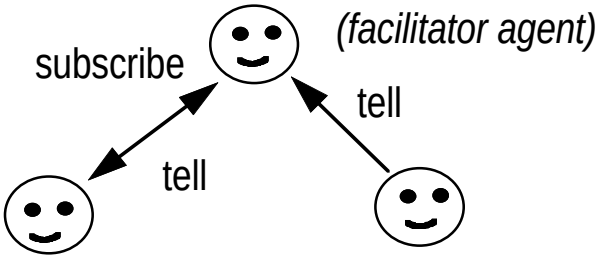
Corkill, Daniel D., and Susan E. Lander, "Organizing Software Agents: The Importance of Design to Effective System Performance," *Object Magazine*, 8:2, April 1998, pp. 41-47.

INTERAGENT COMMUNICATION Á LA KQML

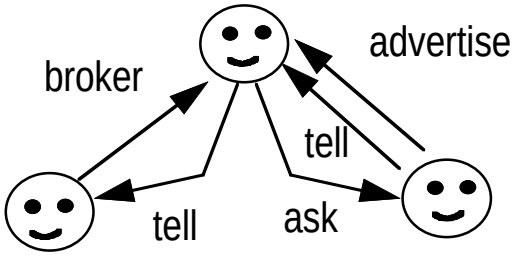
Simple point-to-point



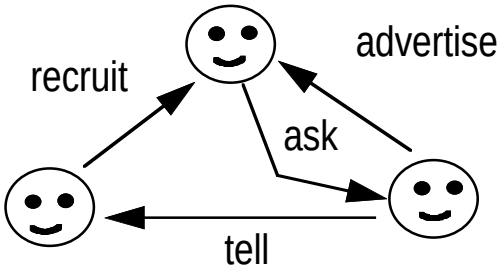
Subscribe



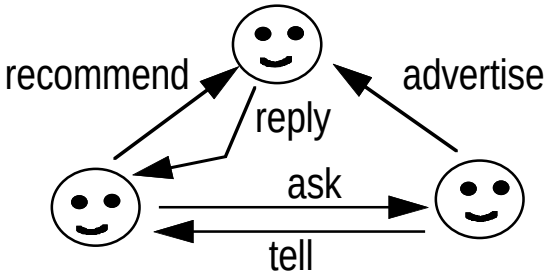
Broker



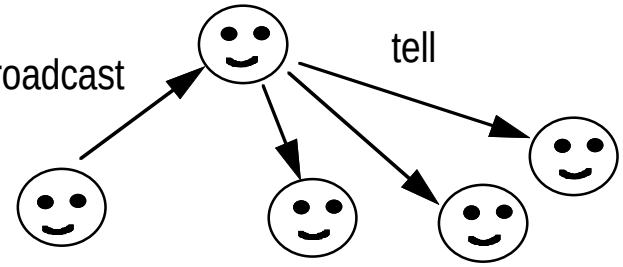
Recruit



Recommend



Broadcast



Finin, Tim, Yannis Labrou, and James Mayfield, "KQML as an Agent Communication Language," *Software Agents*, Jeffrey M. Bradshaw, ed., MIT Press, Cambridge, MA, 1997, pp. 291-316.

A COMMUNICATION ARCHITECTURE FOR SOFTWARE AGENTS INVOLVES:

- *Communication protocols*
  - **Unicast** - sending a packet when there is only one sender process and one specific recipient process.
  - **Broadcast** - sending only one packet and all the hosts in the network recognize and read it.
  - **Multicast** - sending only one packet and all the hosts that were registered as being "of interest" recognize and read it.
- *Application protocols*
  - **Publish/subscribe** - decoupled, asynchronous, may-to-many, event-driven communication.
  - **Request/reply** - decoupled, synchronous, one-to-many, demand-driven communication.
  - **Solicit/response** - asynchronous request/reply.
- *Message routing*
  - **Subject-based**
  - **Content-based**
- *Message properties*
  - **Format repository service**
  - **Self-describing format**
  - **Transformation/translation service**
  - **Message priority**
  - **Message expiration**

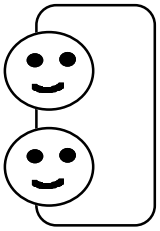
Moreh, Jahan, "Publish & Subscribe: The Power behind Interactive Push Technology," *Distributed Computing*, 1:2, January/February, 1998, pp. 23-27.

AGENTS AND MULTIAGENTS

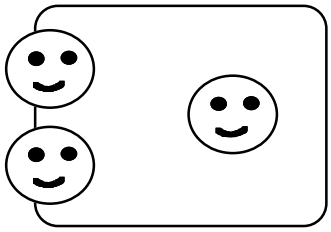
Agents can consist of smaller components, which are themselves agents, and so on.

Option 1:
1 Agent,
Single Boundary

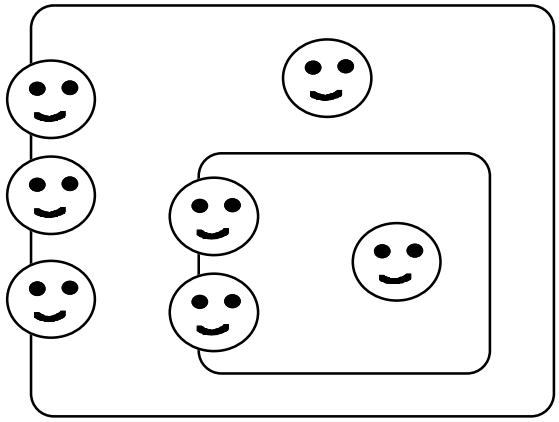
Option 2:
2 Agents,
Single Boundary



Option 3:
Layered
Boundaries



Option 4:
Complex
Aggregate



The behavior of the aggregate is often distinct from the behavior of the parts.

OO AND AGENTS

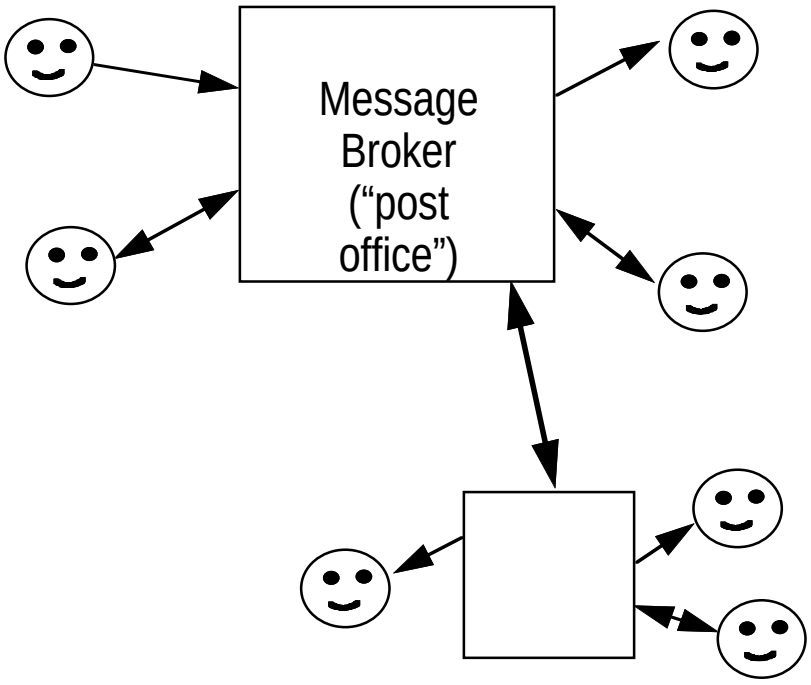
Conventional object orientation:

- ❏ is biased toward class-message-state.
- ❏ is centrally organized; yet some situations require a decentralized and self-organized approach (e.g., flocks of birds and paint stations).
- ❏ depends on external activation of objects as opposed to the continouous and concurrent activity of agents.
- ❏ does not express some business concepts (such as rules, constraints, goals, and roles and responsibilities).
- ❏ OO can be used to enable agent technology.

A COMMUNICATION ARCHITECTURE FOR SOFTWARE AGENTS

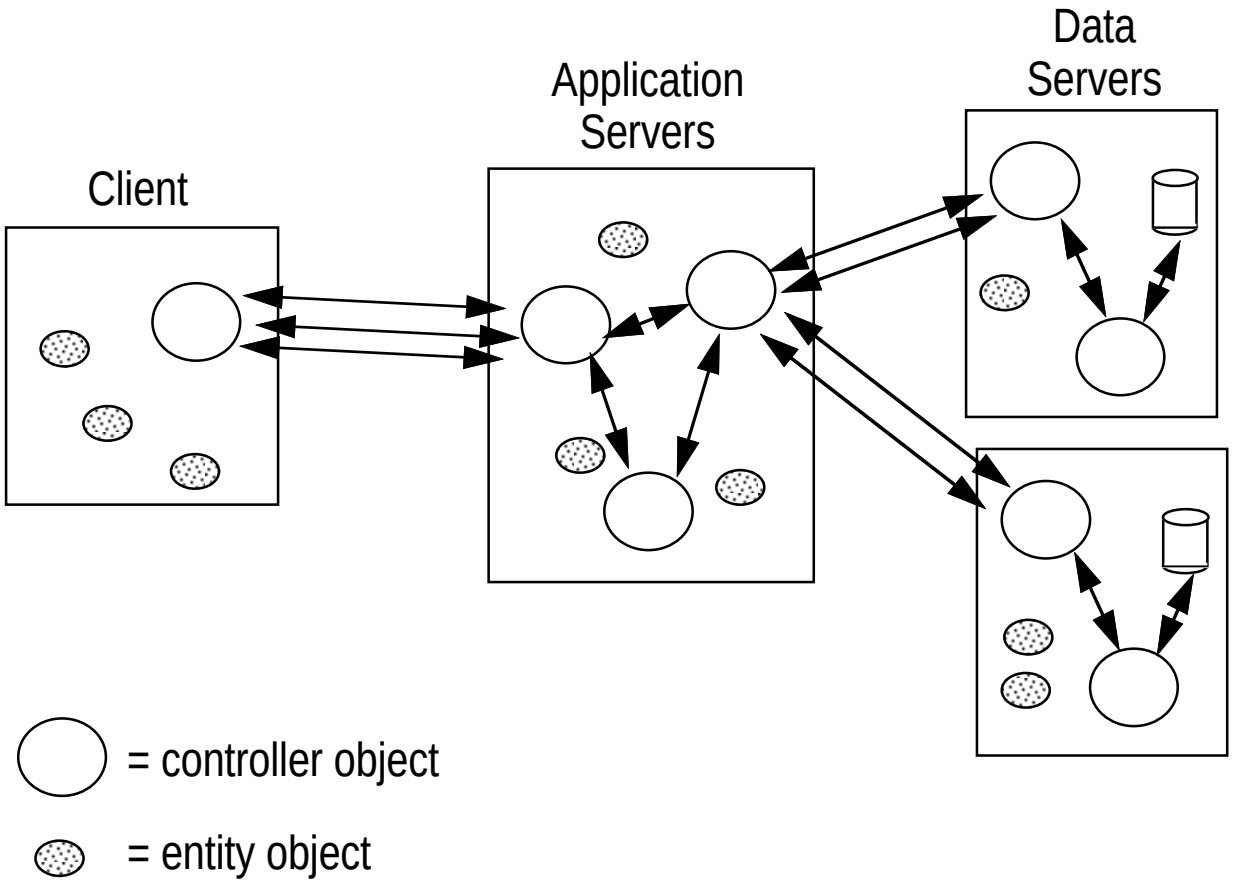
A hub-and-spoke architecture—

with replicated hub



THREE-TIER CLIENT/SERVER

Typical RPC approach:



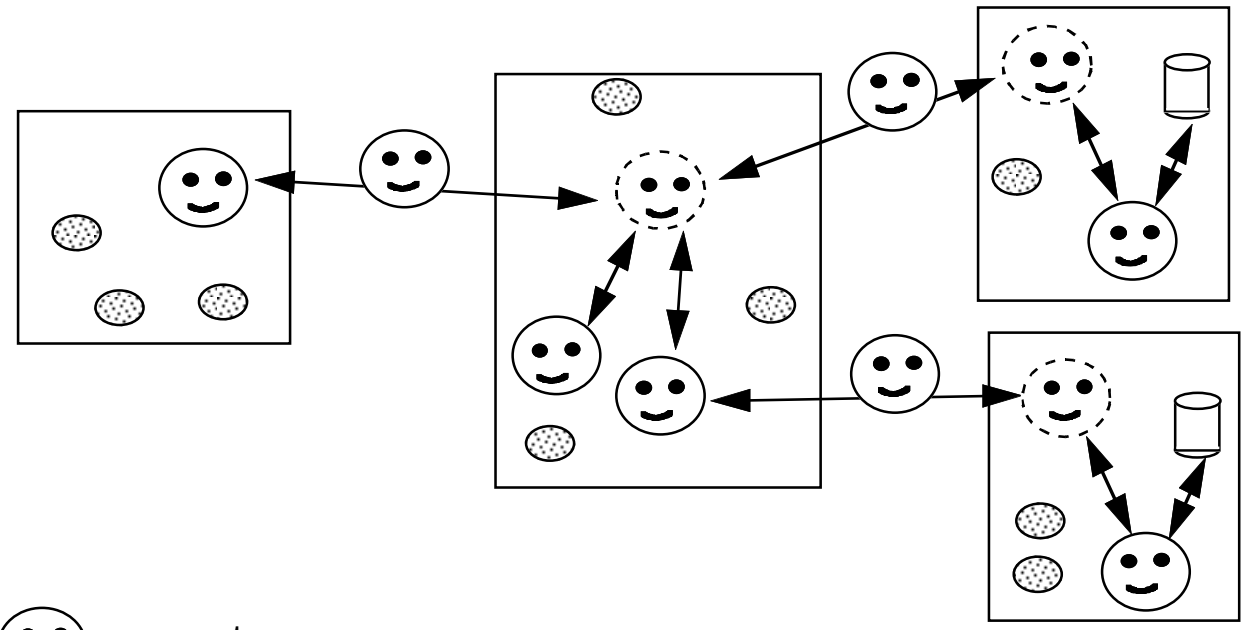
Intermediate servers with "controller objects" are common.

Access if often via an ORB and with SQL calls to DBs.

Graham, Ian, "The Architecture of Agents," *Object Magazine*, 7:7, September 1997, pp. 26-29.

THREE-TIER CLIENT/SERVER

Software-agent approach:



 = agent

 = mobile agent

 = resource

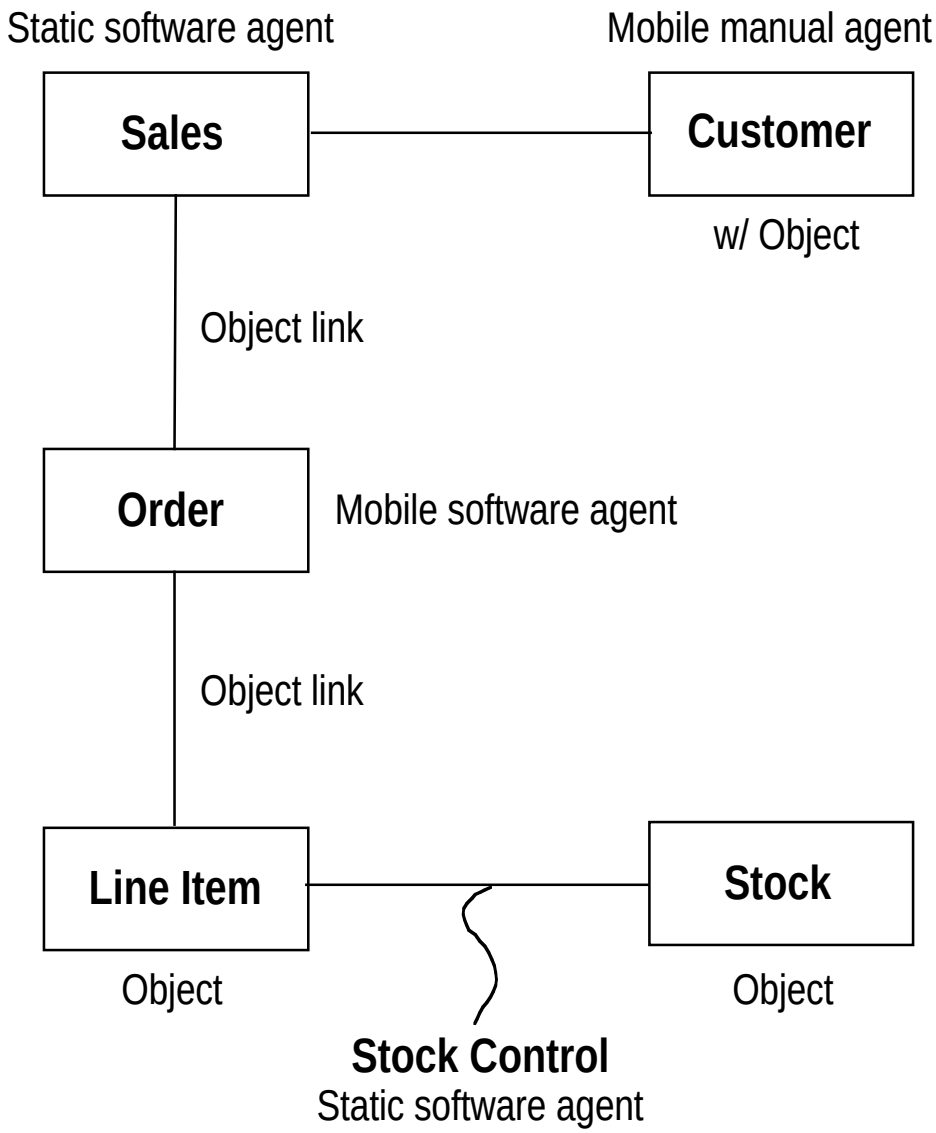
Agents can send a program across a network, rather than using RPCs or SQL. The mobile agent can take over the responsibility of the controller object--without violating encapsulation.

Graham, Ian, "The Architecture of Agents," *Object Magazine*, 7:7, September 1997, pp. 26-29.

WHAT GETS AGENTED?

One technique: start with a conceptual-level class diagram, and identify which objects and links are candidates for agent-hood.

An example with one possible sotution.



ADAPTIVE AGENT

An agent that responds to its environment

Three primary ways of adapting:

- **reaction**- a direct, predetermined response to a particular event or environmental signal.

- **learning**- change that occurs during the lifetime of an agent.

- **evolution**- change that occurs over succesive generations of agents.

ADAPTING BY REACTION

A direct, predetermined response to a particular event or environmental signal

Typically expressed in the form:

WHEN event, IF conditon(s), THEN action.

- ❏ thermostats
- ❏ robotic sensors that can detect the presence of a nearby wall and activate a device for avoiding it
- ❏ washing machines and vacuum cleaners that use fuzzy logic

A more advanced form uses a set of rules to perform inferencing.

- ❏ patient diagnosis
- ❏ bulletin board or web foraging agents
- ❏ data mining

(Reflex versus instinct)

ADAPTING BY LEARNING

Change that occurs during the lifetime of an agent

Typical kinds of techniques:

- ❏ credit assignment
- ❏ Bayesian (or probablistic) rules
- ❏ neural networks
- ❏ classifier rules
- ❏ problem-specific structures

ADAPTING BY EVOLUTION

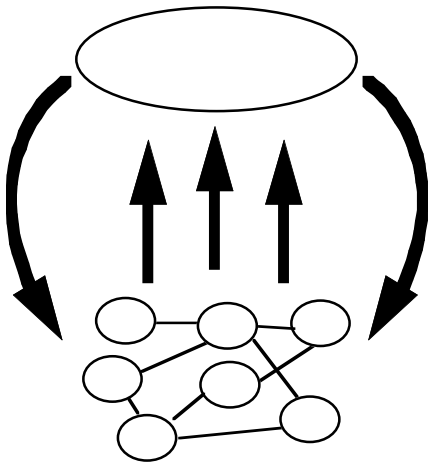
Change that occurs over successive generations of agents

Typical kinds of strategies:

- ❏ natural selection, i.e., “survival of the fittest”
- ❏ Darwinian versus Lamarckian evolution (e.g., genotype and phenotype)
- ❏ differentiation into ecosystem roles
- ❏ competition (e.g., increasing returns)
- ❏ cooperation (e.g., multiagent composition)
- ❏ coevolutionary arms races
- ❏ cultural transmission (e.g., Richard Dawkins’ “memes”)

EMERGENCE

The existence of a coherent pattern that arises out of the collective behavior and interactions among simpler objects



- ❏ Agents organize into a larger structure that is greater than the sum of its parts.
- ❏ If a cluster is coherent and stable enough, it can usually serve as a building block for some larger cluster.
- ❏ At each level, new emergent structures can form and engage in behaviors that can lead to further emergence.
- ❏ Building blocks at one level, then, can combine to form building block at a higher level.

Complexity is the science of emergence.

CONCLUSION

- Agents are an evolution, not a revolution.
- But, our usage of them will be revolutionary.
- Yet, our current usage is typically limited to
  - agents acting individually.
  - person service agents.
  - Internet-based applications.
- A major opportunity for agents exists for
  - decision making and design.
  - solving large combinatorial optimization problems.
  - detecting and responding to opportunities and challenges in complex dynamic situations.
  - . . .