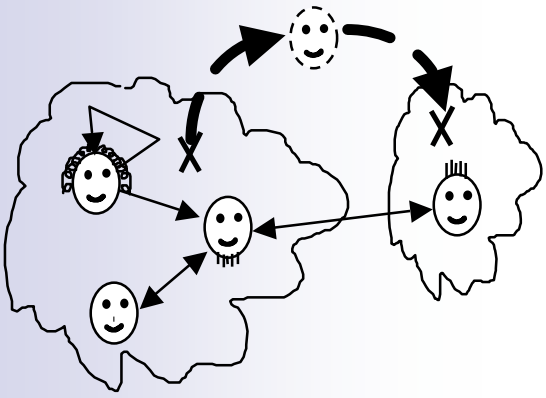
A Tutorial for NODE 2003

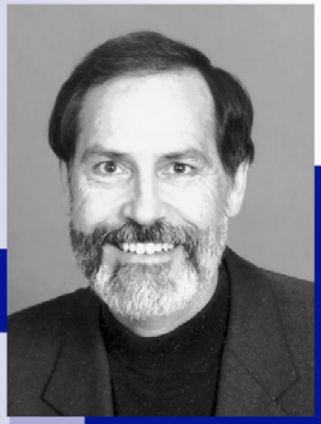
Beyond Objects: Unleashing the Power of Adaptive Agents



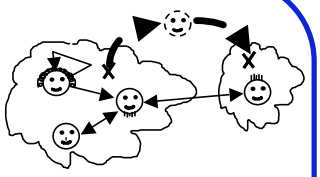
Presenter

James Odell

James Odell Associates
3646 W. Huron River Drive
Ann Arbor, Michigan 48103-9489 USA
telephone: +1 (734) 994-0833
email: email@jamesodell.com
website: www.jamesodell.com



A Tutorial for NODE 2003



Beyond Objects:
Unleashing the Power of Adaptive Agents

Chair, OMG A&D "UML" Task Force
Chair, FIPA Agent Modeling Technical Committee
OMG Certified UML Advanced Professional

James Odell
James Odell Associates
3646 W. Huron River Drive
Ann Arbor, Michigan 48103-9489 USA
telephone: +1 (734) 994-0833
email: email@jamesodell.com
website: www.jamesodell.com

A FLOCK IS NOT A BIRD



COORDINATED WITHOUT A COORDINATOR

Such as,

- the photorealistic imagery of bat swarms used in *Batman Returns* and *Cliffhanger*
- the wildebeest stampede in *The Lion King*,
- ant colonies, stadium crowds, highway traffic, market economies, and immune systems.

Craig Reynolds (formerly from Symbolics Corp. and DreamWorks); http://www.red3d.com/cwr/boids/



StarLogo is described in *Turtles, Termites, and Traffic Jams* by Mitchel Resnick.
Software can be downloaded from http://education.mit.edu/starlogo/.

COMPLEX ADAPTIVE SYSTEMS

A way of thinking about multiagent systems and their collective behavior

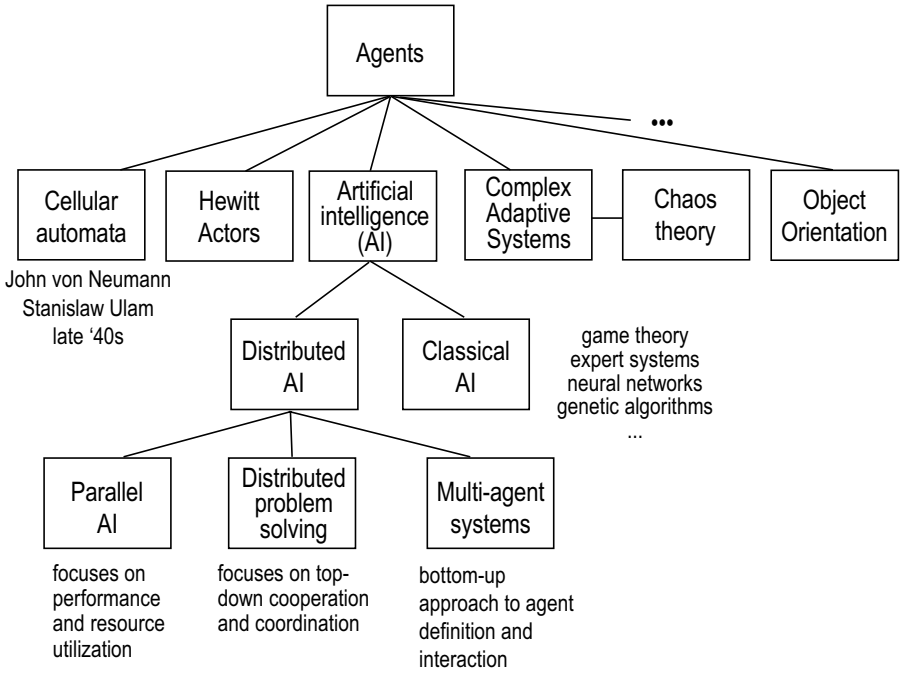
The study of the behavior of collections of simple units—or agents
e.g., atoms, molecules, neurons, people, that have the potential to adapt
and can give rise to coherent collective behavior.

Also,

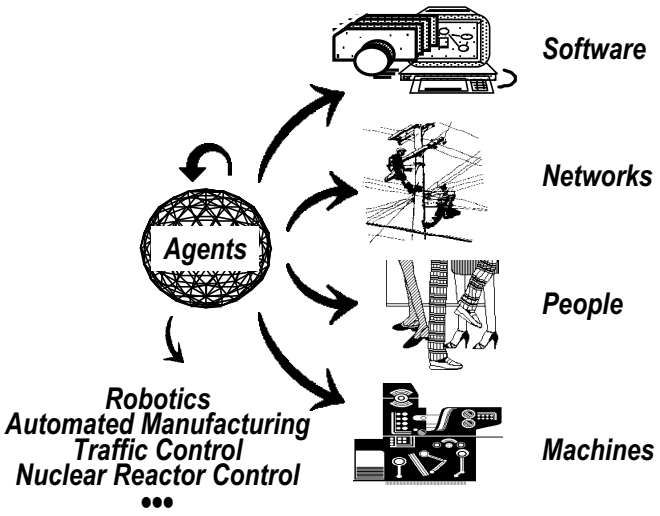
System (science)	Typical Mechanisms
Nucleus (physics)	Quarks, gluons
Atom (physics)	Protons, neutrons, electrons
Molecule (chemistry)	Bonds, active sites, mass action
Organelle (microbiology)	Enzymes, membranes, transport
Cell (biology)	Mitosis, meiosis, genetic operators
Multicellular organism (biology)	Morphogenesis, reproduction
Social group (biology)	Individuals, social relationships
Ecosystem (ecology)	Symbiosis, predation, mimicry

And, it can provide scalability!

WHAT ARE THE ROOTS OF AGENT TECHNOLOGY?



AGENTS CAN BE IMPLEMENTED IN MANY WAYS



HOW ARE WE BEGINNING TO USE AGENTS?

In general:

- **Personal use**
- **Interest matching**
- **Network and system management-** load balancing, failure anticipation, problem analysis, and grid management.
- **Information-** synthesis, decision, and logistics support
- **Process control-** ensure activities are carried out (e.g., EAI, SOA)
- **Business process management-** capable of providing a variety of services, e.g., multiagent supply chain systems
- **e-Commerce-** with buying, brokering, bidding, and selling agents
- **Product design-** designing components and subsystems of a complex product

Some application areas:

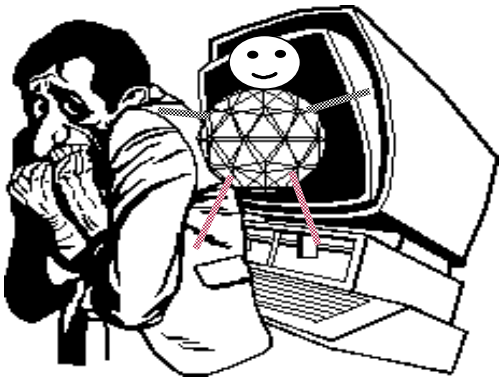
- Manufacturing
- Utilities (electric, gas, telephone)
- Information
- Retail
- Finance (banks, insurance, stock market)
- Health
- Government (military, NASA)
- Air traffic control
- Media
- E-Mail
- Web personalization
- . . .

TUTORIAL ROADMAP

- **What is an agent?**
- **Agent anatomy**
- **Agent communication**
- **What gets agented?**
- **Adaptation**
- **Emergence**
- **Order and Chaos**
- **Decentralization versus centralization**
- **Designing for emergence**
- **Conclusion**

… and punctuated by lots of examples and animations

AGENTS: WHAT ARE THEY?



AGENT

"Something that acts" — Webster's Dictionary

Some common properties of agents:

- **Autonomous -** is capable acting without direct external intervention.
- **Interactive -** communicates with the environment and other agents.
- **Adaptive -** capable of responding to other agents and/or its environment
- **Sociable -** interaction that is marked by friendliness or pleasantness
- **Mobile -** able to transport itself from one environment to another.
- **Proxy -** may act on behalf of someone or something.
- **Proactive -** goal-oriented, purposeful; does not simply react..
- **Intelligent -** state is formalized (i.e., beliefs, goals, plans, assumptions) and uses symbolic language.
- **Rational -** able to act based on internal goals and knowledge.
- **Unpredictable -** able to act in ways that are not fully predictable.
- **Temporally continuous -** is a continuously running process.
- **Credible** - believable personality and emotional state.
- **Transparent and accountable -** must be transparent when required, yet must provide a log of its activities
- **Coordinative -** able to perform some activity in a shared environment with other agents
- **Cooperative -** able to coordinate with other agents to achieve a common purpose.
- **Competitive -** able to coordinate with other agents where the success of one implies the failure of another.
- **Trustworthy -** adheres to Laws of Robotics and is truthful.

TWO ASPECTS OF AUTONOMY

Unpredictable Autonomy

- Can say "No"
- Predictable vs. Unpredictable
- Determined by external observer

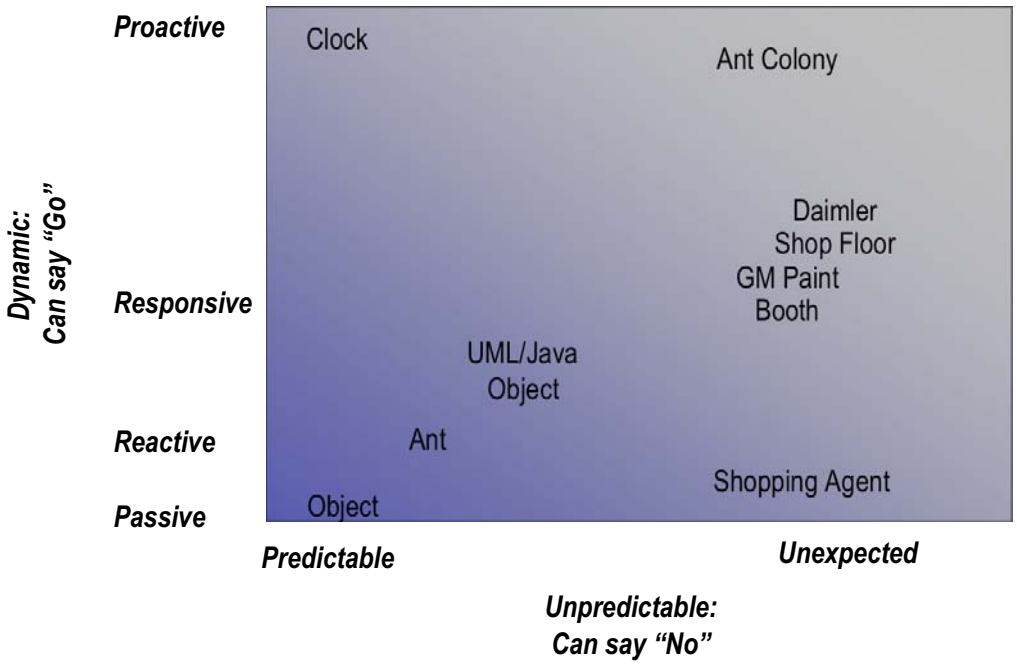




Dynamic Autonomy

- Can say "Go"
- Reactive vs. proactive
- Determined by agent's internal structure

TWO ASPECTS OF AUTONOMY



ASPECTS OF INTERACTION AND ADAPTATION

Interaction

- Communication
- Coordination
- Cooperation
- Competition
- . . .

Adaptation

- Reaction
- Reasoning
- Learning
- Evolution
- . . .

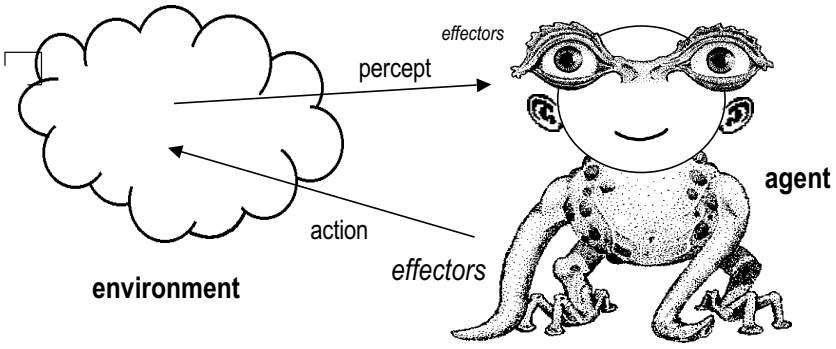
AGENT

Basic Working Definition

An autonomous entity that can adapt and interact with its environment.

In other words, it is anything that can be viewed as:

- perceiving its environment though sensors, and
- acting upon an environment with effectors.



To be useful, agents should—to some extent—be *autonomous*, *adaptive*, and *communicative*.

SOFTWARE TECHNOLOGY AND AGENTS

Software history is one of increasing localization and encapsulation.

	Monolithic Programming	Modular Programming	Object-Oriented Programming	Agent Programming
Unit Behavior	Nonmodular	Modular	Modular	Modular
Unit State	External	External	Internal	Internal
Unit Invocation	External	External (CALLed)	External (message)	Internal (rules, goals)

Parunak, H. Van Dyke, *Autonomous Agent Architectures: A Non-technical Introduction*, Industrial Technology Institute, 10/13/95.

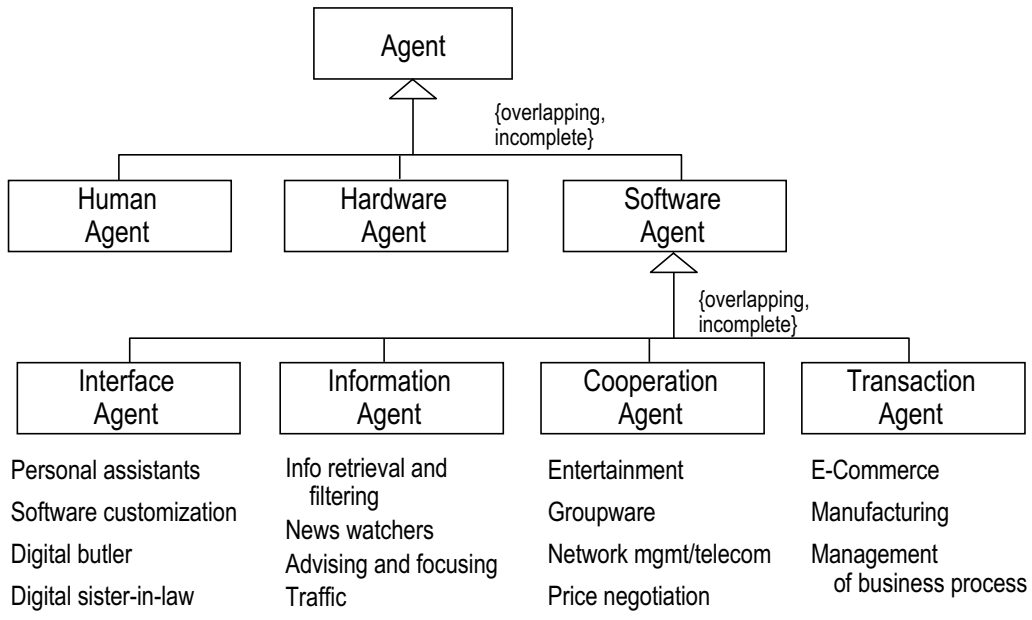
OO AND AGENTS

Conventional object orientation:

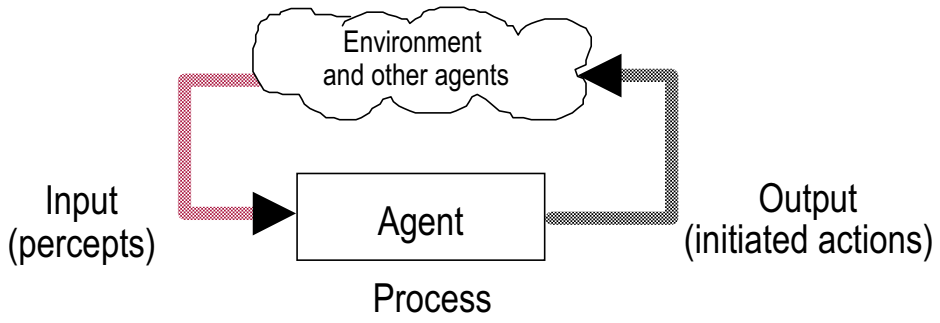
- is biased toward class-message-state.
- is centrally organized; yet some situations require a decentralized and self-organized approach (e.g., flocks of birds and paint stations).
- depends on external activation of objects as opposed to the continuous and concurrent activity of agents.
- does not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

AGENT CATEGORIES

A common taxonomy



AGENT AS A BLACK BOX

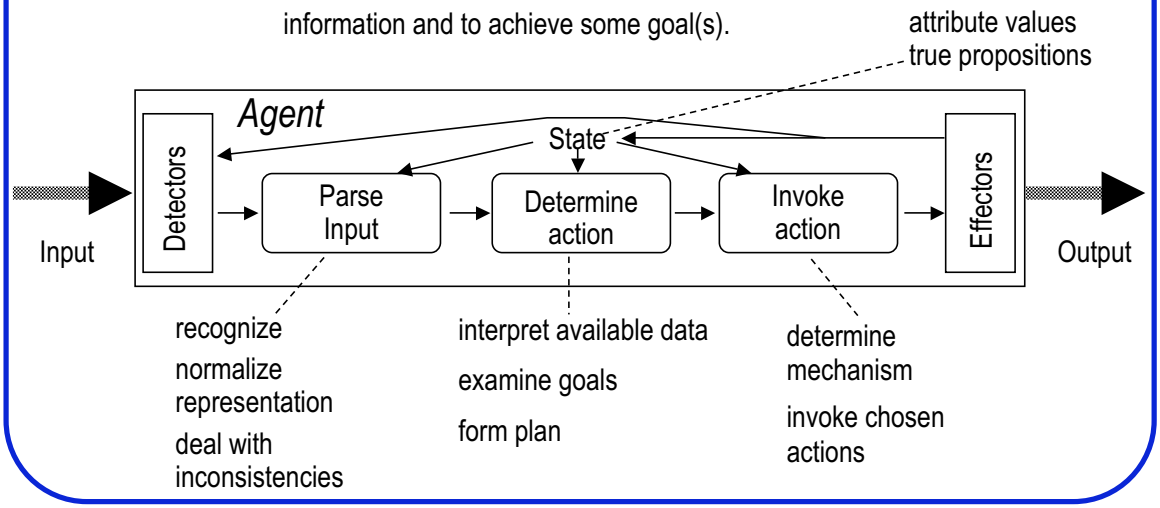


- **Input -** can be whatever the agent "perceives": assertions, queries, commands, state changes— that is, it is richer than OO messages.
- **Process -** interprets the perceived information, determines its action, and invokes it.
- **Output -** the initiated action, such as a change or communication to its world.

AGENT PROCESSING OVERVIEW

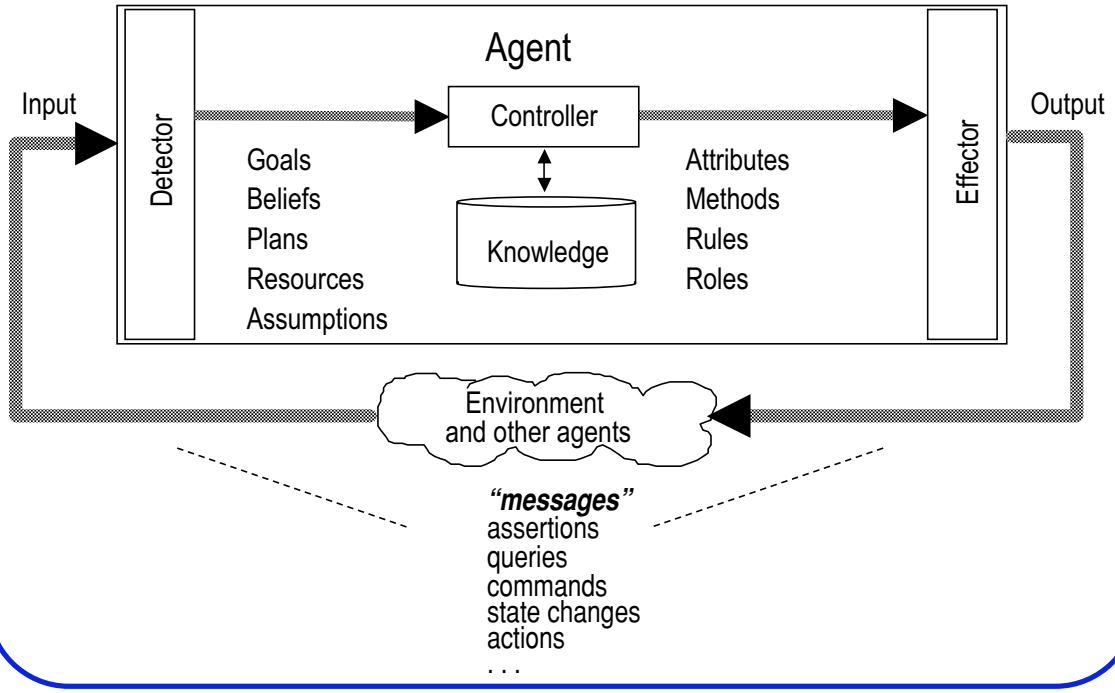
The primary purpose of an agent:

- is not just to interact with the environment,
- but rather to process and interpret the perceived information and to achieve some goal(s).

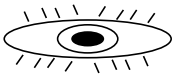


HIGH-LEVEL AGENT VIEW

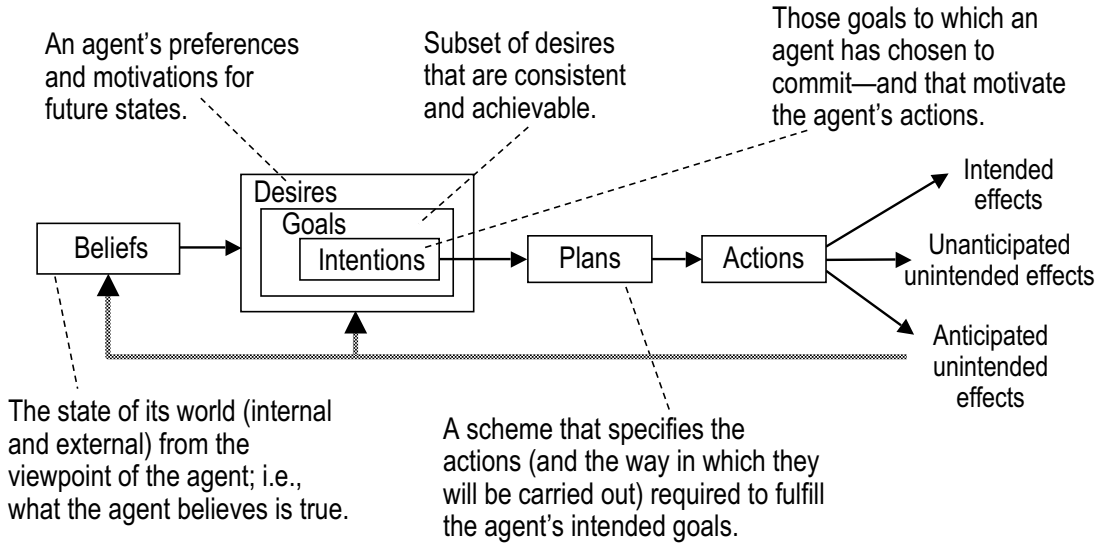
Possible agent anatomy and context



BDI AGENTS

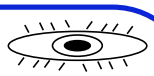


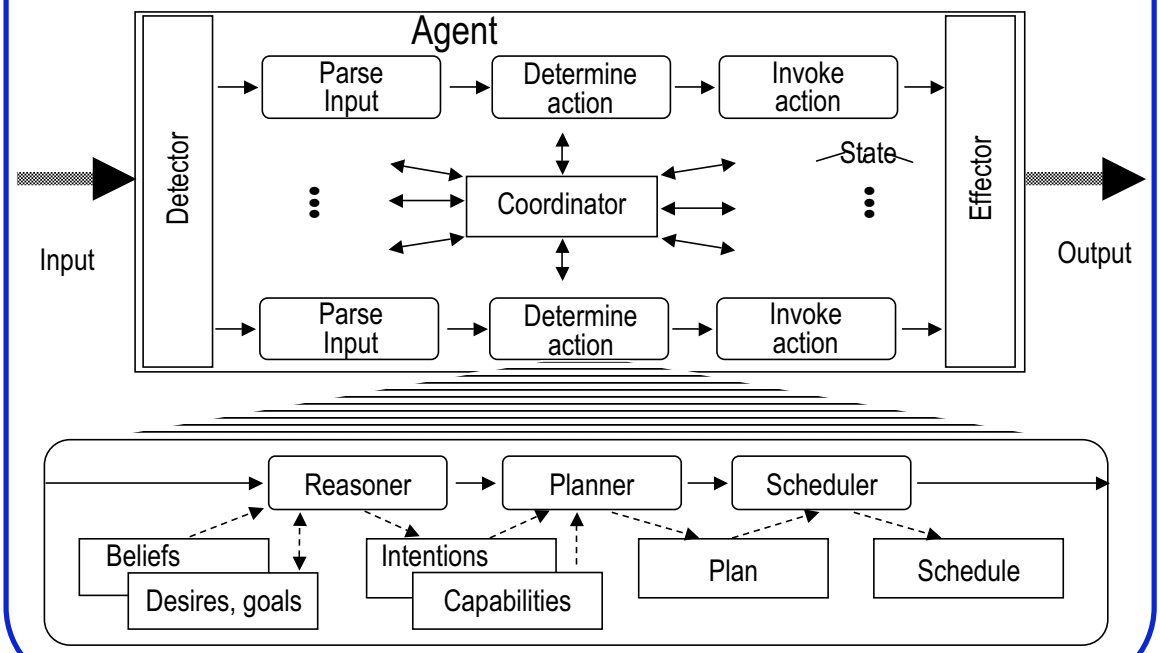
"Mental states" guide the agent selection of appropriate actions.



*BDI = belief, desire, intention

AT ONE EXTREME: BDI AGENTS WITH BELIEFS, DESIRE, INTENTIONS

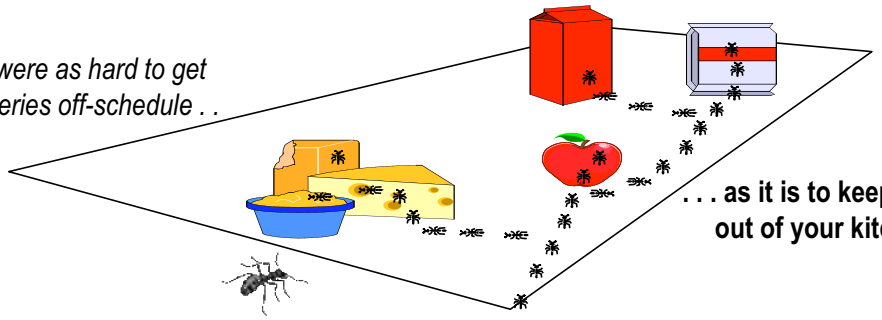




AT ANOTHER EXTREME…

**if we built systems with ant-like simplicity,
we could create something very successful.**

What if it were as hard to get your deliveries off-schedule . . .



. . . as it is to keep ants out of your kitchen?

- Ants have been very successful for a very long time (7 million years).
- Estimated 10% of the Earth's animal biomass is composed of ants. (Much more than humans)
- An ant has about 250,000 brain cells; humans have about 10 billion. (Simple rules)
- Ants can lift 20 times their own body weight. (They may be small, but they're more efficient.)
- If a person could run as fast for his size as an ant, he would match the pace of a race horse.

Simple rules, high fault tolerance—and the colony thrives. A great first-system model.

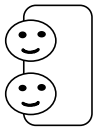
AGENTS AND AGGREGATE AGENTS

Communication can be with individuals as well as groups via agents defining the interface.

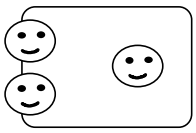
Option 1:
1 Agent,
Single Boundary



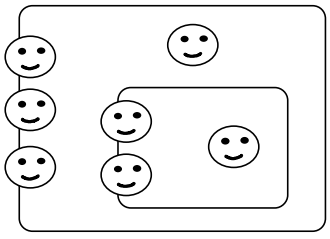
Option 2:
2 Agents,
Single Boundary



Option 3:
Layered
Boundaries



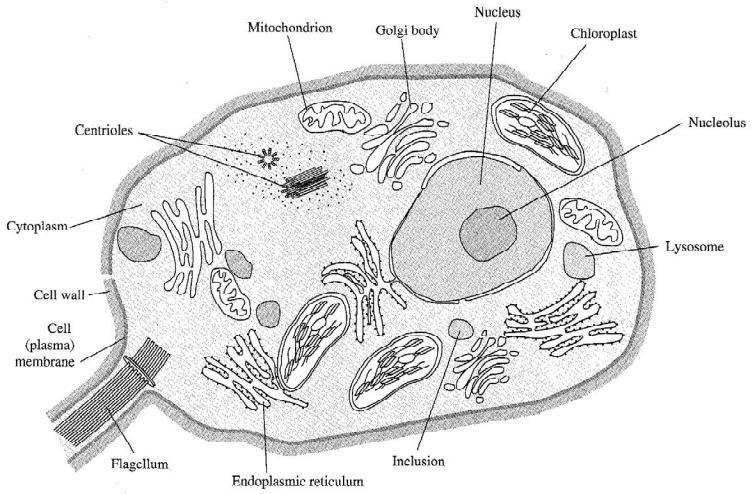
Option 4:
Complex
Aggregate



- The behavior of the aggregate is distinct from the behavior of the parts.
- The "membrane" agents provide the interface definition, i.e., the "cell" is encapsulated.

AGENTS AND AGGREGATE AGENTS

Using nature as an analogy

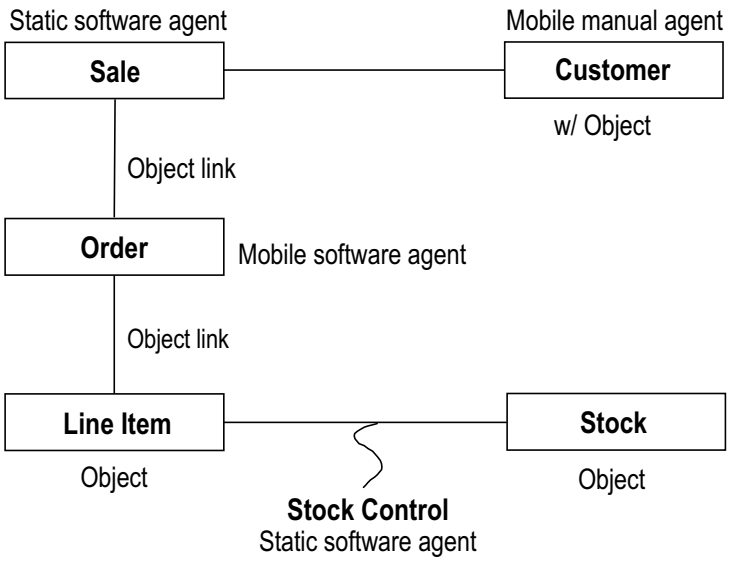


- symbiosis
- parasitism
- mimicry
- niche formation
- speciation
- sexual and asexual reproduction
- eukaryotic versus prokaryotic cells
- Darwinian evolution
- Lamarckian evolution
- neuron-based systems
- ...

WHAT GETS AGENTED?

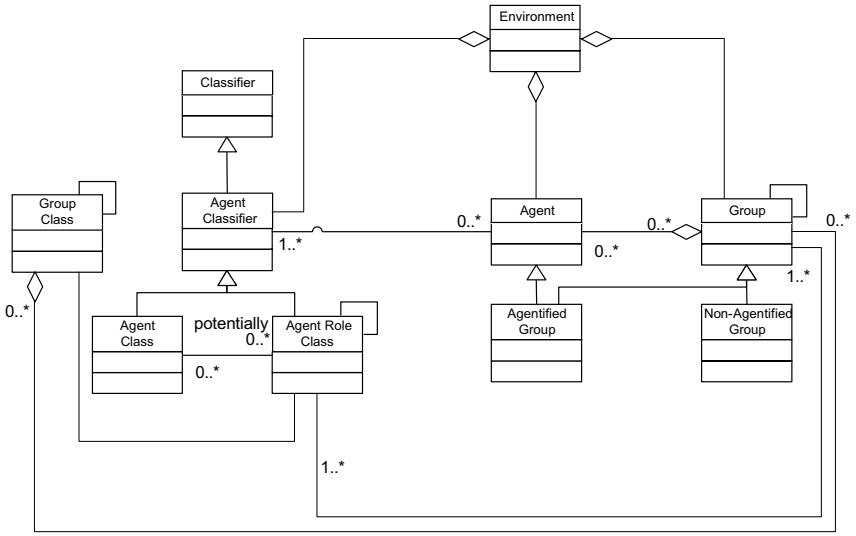
One technique: start with a conceptual-level class diagram and identify which objects and links are candidates for agent-hood.

An example with one possible solution.



AGENT METAMODEL

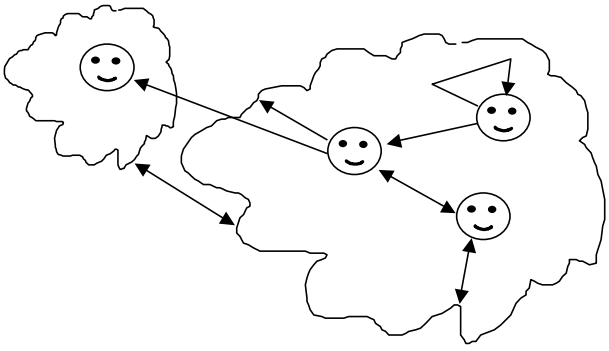
Current thinking on the metamodel required to support agents.



AGENT COMMUNICATION

Agents can communicate with other agents—

as well as with their environment.



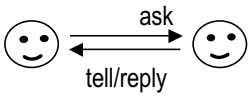


In fact, the environment itself can be

treated as an agent, when appropriate.

AGENT COMMUNICATION LANGUAGE (ACL)

Simple point-to-point



KQML examples:

```
(ask-one
:sender joe
:content (PRICE IBM ?price)
:receiver stock-server
:reply-with ibm-stock
:language LPROLOG
:ontology NYSE-TICKS)
```

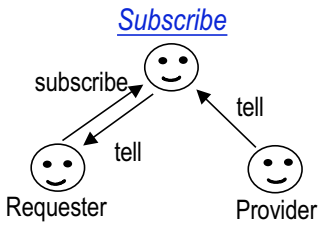
```
(tell
:sender stock-server
:content (PRICE IBM 140)
:receiver joe
:in-reply-to ibm-stock
:language LPROLOG
:ontology NYSE-TICKS)
```

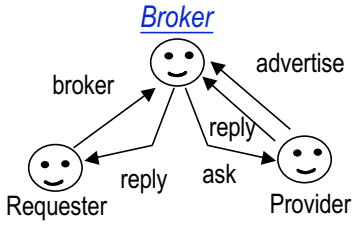
```
(ask-all
:content "price(IBM, [?price, ?time])"
:receiver stock-server
:language standard_prolog
:ontology NYSE-TICKS)
```

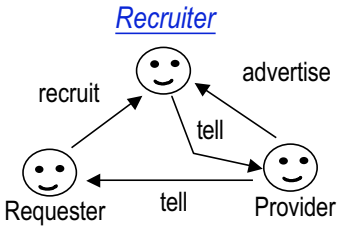
```
(stream-all
;;?VL is a large set of symbols
:content (PRICE ?VL ?price))
```

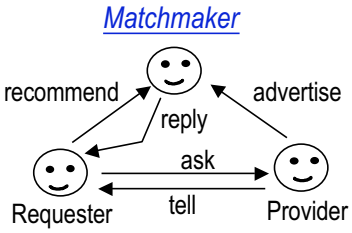

INTERAGENT COMMUNICATION PROTOCOLS USING KQML

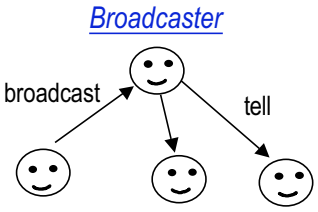
With facilitator agents





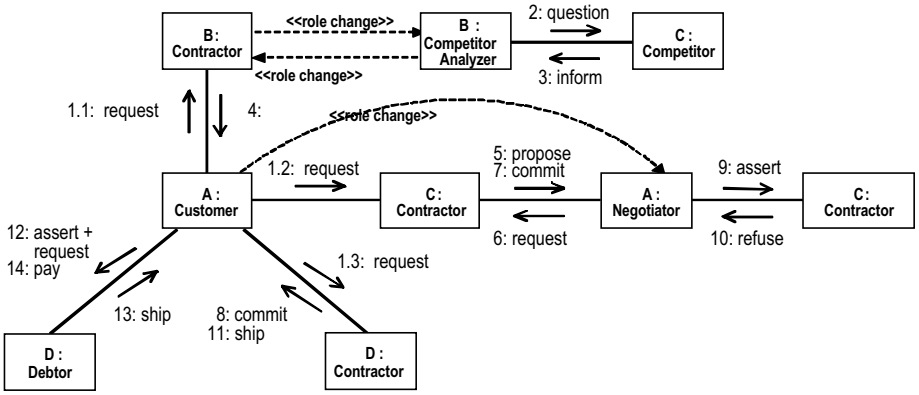






INTERAGENT COMMUNICATION

Role-based architecture

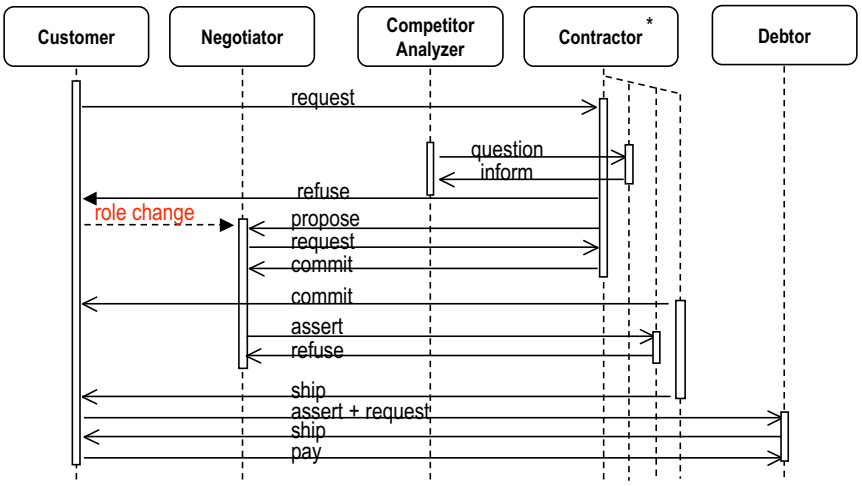


Agents in roles using a UML communication diagram (previously known in UML as the collaboration diagram).

Also see http://www.auml.org

INTERAGENT COMMUNICATION

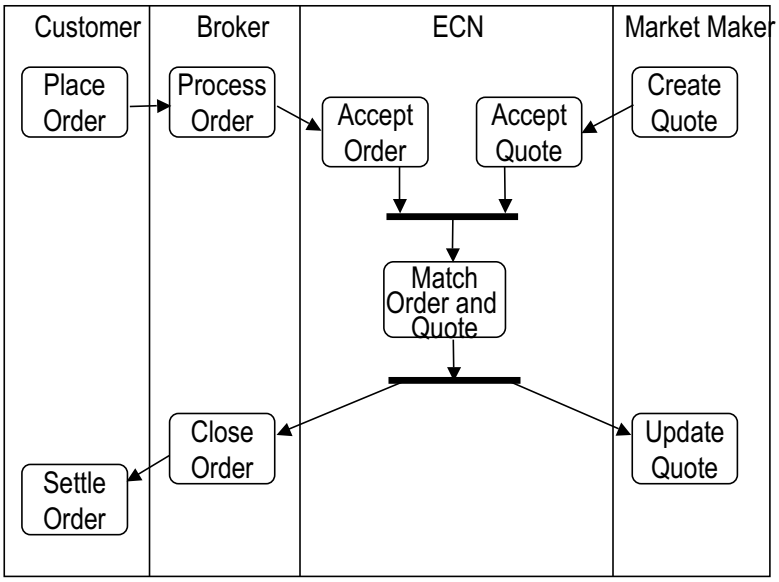
Role-based architecture



Agents in roles using a UML sequence diagram.

INTERAGENT COMMUNICATION

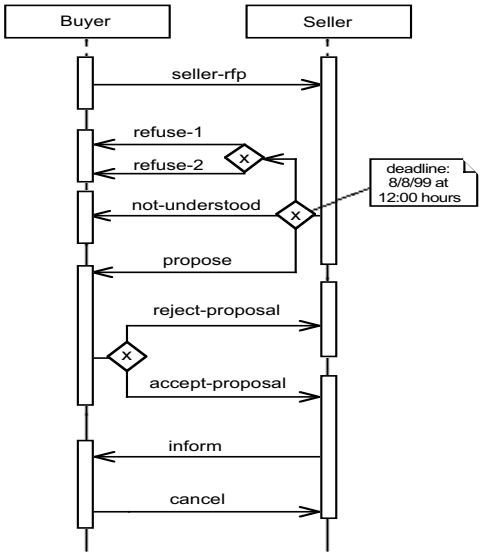
Role-based architecture



Agents in roles using a UML activity diagram.

INTERACTION PROTOCOLS

a communication pattern as an allowed sequence of messages between agents.



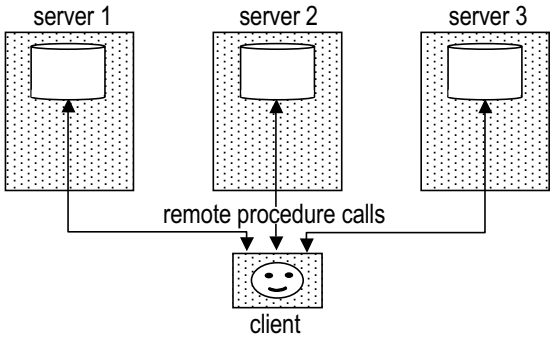
FIPA INTERACTION PROTOCOLS

- Request
- Query
- Request When
- Contract Net (Negotiation)
- Iterated Contract Net
- Auction English
- Auction Dutch
- Brokering
- Recruiting
- Subscribe
- Propose

(A partial list.)

STATIONARY AGENTS

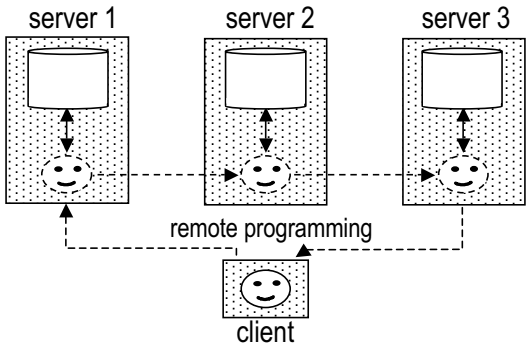
Primarily uses Remote Procedure Call (RPC) technique for remote work.



- When an agent wants to use the services of another agent, a message (or request) conveys the intention to invoke a specific operation.
- The operation is then executed and the results (or reply) is returned to the requesting agent.
- Standard client-server protocols.

MOBILE AGENTS

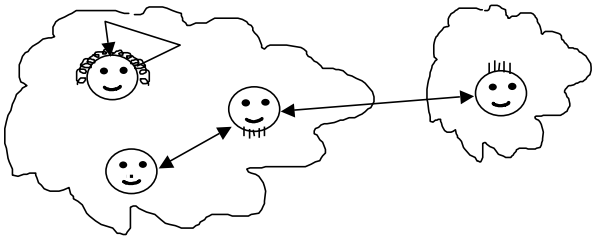
Primarily uses Remote Programming (RP) technique for remote work.



- All structural and behavioral properties of the agent must be transferred during migration.
- Environmental differences must be changed or accommodated.
- The big issues:
  - How much time it takes to prepare for migration
  - How much data is actually transferred
  - The performance of the communication
  - Server-side burden

SINGLE VS. MULTIAGENT AGENT SYSTEM (MAS) APPROACHES

Single agents have their use, but using many agents to solve problems can also be useful.

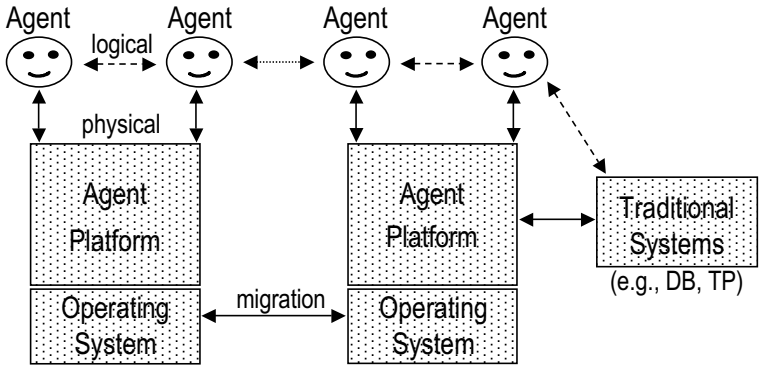


- One agent *could* be constructed that does everything—but would be a bottleneck for speed, reliability, etc.
- Dividing the functionality among many agents provides modularity, flexibility, modifiability, and extensibility.
- Support distributed processing and problem solving
- Specialized knowledge is often not available from a single agent.
- Knowledge is typically spread over various agents.
- Single-agent systems are much simpler because they don't deal with cooperation, negotiation, etc.

AGENT ARCHITECTURE

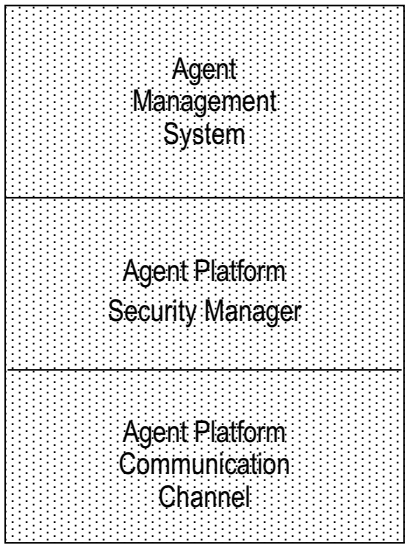
Agent communication can be made in two ways:

- Directly with each other which
  - ✔ provides flexibility and freedom.
  - ✘ bypasses control and security.
- Through base software (preferred) which
  - ✔ resolves control and security problems.
  - ✘ requires "logical" communications which are physically resolved via the base software.



AGENT ARCHITECTURE

Three layer FIPA Agent Platform



Execution and monitoring of active agents
Basic functionality (API)
- Identification
- Directory Services
- Registration
- Query/Search
- Negotiations
- Mobility

Secure transfer of messages and objects
Secure protocols
Data encryption
Digital signature
Firewalls

Provision of base communication functions
Protocols, document formats
RPC, remote programming
Remote method invocation
Object serialization

"FIPA Abstract Architecture Specification," FIPA Document PC00001, 2000.

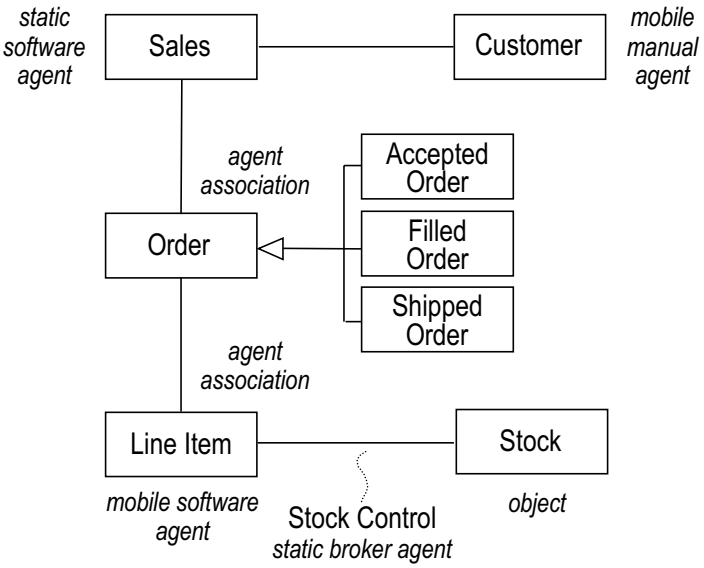
WHAT GETS AGENTED?
Common approaches

- **Type (or role) based** - identify which types and relationships are candidates for agent-hood.
- **Vertical vs. horizontal** - identify the degree and distribution of functional expertise that an agent should possess.
- **Workflow based** - well-defined activities can be become the basis for agent assignment
- **Hierarchical vs. network** - identify the interaction structure that agents should use.
- **Redundant vs. disjoint** - identify agents based on efficiency and robustness.
- **Incomplete, inaccurate, and inconsistent information**.
- **Global coherence with decentralized control**.

TYPE (OR ROLE) BASED

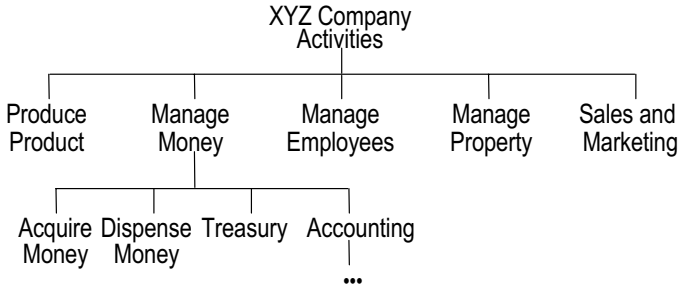
Start with a conceptual-level class diagram and identify which objects and links are candidates for agent-hood.

An example with one possible solution.



HORIZONTAL VS. VERTICAL SPECIALIZATION

- **Horizontal Specialization** - each agent is an "expert" at some basic business function (e.g., finance, accounts payable, check issuance). A given task or problem is routed to a number of appropriate experts, as necessary.
  - ✓ Effective, orderly organization of expertise
  - ✓ Resource life cycles are often "natural" breakdowns
  - ✗ Fragile (unless with redundancy)
  - ✗ "Fat" agents
  - ✗ Distributed inter-agent communication



HORIZONTAL VS. VERTICAL SPECIALIZATION

- **Vertical Specialization** - each agent possesses most or all the required expertise to perform a given task or solve a given problem. Here, the agent is an expert on certain situations in its spatial neighborhood.
  - ✓ Minimizes distributed communication
  - ✗ Fragile
  - ✗ "Fat" agents

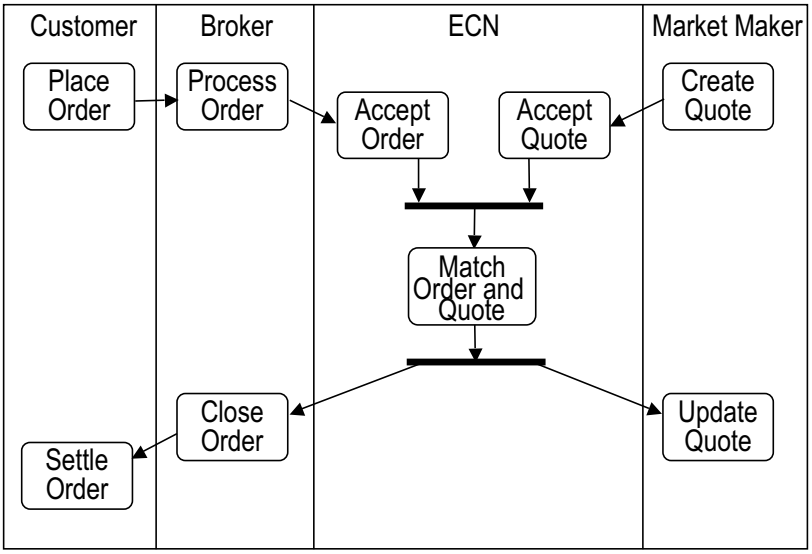
For example:



- **Mixed and hybrid specialization** - a mixture of horizontal and vertical specialization

WORKFLOW BASED

Start with a conceptual-level activity diagram and identify which roles are candidates for agent-hood.



Example using agents in roles using a UML activity graph.

HIERACHICAL VS. NETWORK

- **Hierarchical structure** - an agent can act as a single point of control for other agents.
  - ✔ Works well when control or results needs to be concentrated at one point.
  - ✗ Is sensitive to the loss of a high-level agent in hierarchy.
- **Network structure** - control is distributed among many agents.
  - ✔ More robust to the loss of an agent.
  - ✗ Increased communication and control problems.
- **Hybrid structure** - a mixture of hierarchical and network structures.

REDUNDANT VS. DISJOINT ACTIVITIES

- **Redundant activities** - multiple agents performing the same activity for the same purpose.
  - ✔ Good insurance against agent failure.
  - ✔ Greater probability of successfully accomplishing the required activity.
  - ✗ Consumes more resources.
- **Disjoint activities** - agents do not perform redundantly.
  - ✔ Efficient use of resources.
  - ✗ Leaves the system severely degraded if a crucial activity is lost due to agent failure.
- **Hybrid activities** - Have just the crucial activities redundantly performed as insurance against agent failure.

INCOMPLETE, INCONSISTENT, AND INACCURATE INFORMATION

- Agents cannot (or do not) always have all the relevant information needed for their local computations and control decisions.
- Each agent may possess enough information to perform a portion of a function, but not enough to perform any function completely.
- Distributed information can often result in unacceptable delays, so action must be taken based on what *is* available.
- The information can be incomplete, inaccurate, inconsistent, or contradictory.
- This is particularly a problem for distributed systems.

GLOBAL COHERENCE WITH DECENTRALIZED CONTROL

In distributed or highly populated agent systems:

- A global "controller" agent is not an option, because it would be a bottleneck and the single point of failure.
- Instead, each agent must be able to direct its own activities in concert with other agents—based on incomplete, inaccurate, and inconsistent information.

In developing such systems, the following must be addressed:

- Connectivity - Agents must interact in a manner that permits the covering activities to be integrated into the overall solution.
- Coverage - Any necessary portion of the overall problem must be included in the activities of at least one agent.
- Capability - Coverage and connectivity must be achievable with available resources.

COMMON APPROACHES

- **Type (or role) based** - identify which types and relationships are candidates for agent-hood.
- **Vertical vs. horizontal** - identify the degree and distribution of functional expertise that an agent should possess.
- **Workflow based** - well-defined activities can be become the basis for agent assignment
- **Hierarchical vs. network** - identify the interaction structure that agents should use.
- **Redundant vs. disjoint** - identify agents based on efficiency and robustness.
- **Incomplete, inaccurate, and inconsistent information**.
- **Global coherence with decentralized control**.

ADAPTATION



ADAPTIVE AGENT

An agent that responds to its environment

Four primary ways of adapting:

- **reaction** - a direct, predetermined response to a particular event or an environmental signal.
- **reasoning** - ability to make inferences.
- **learning** - change that occurs during the lifetime of an agent.
- **evolution** - change that occurs over successive generations of agents.

} Self-adaptive

ADAPTING BY REACTION

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

Typically expressed in the form—

WHEN event, IF condition(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

ADAPTING BY REASONING

A reactive response that uses inference rules

A more advanced form of reactive adaptation using a set of rules to perform inferencing:

Typically *chains* of rules in the form—

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

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

ADAPTING BY LEARNING

Change that occurs during the lifetime of an agent

Typical kinds of techniques:

- credit assignment
- Bayesian (or probabilistic) rules
- neural networks
- classifier rules
- problem-specific structures
- Q-learning

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
- differentiation into ecosystem roles
- competition (e.g., increasing returns)
- cooperation (e.g., multiagent composition)
- co-evolutionary arms races
- cultural transmission (e.g., Richard Dawkins' "memes")

ADAPTIVE AGENT

Four primary ways of adapting

Summary

- Reaction (minimum requirement)
- Reasoning
- Learning
- Evolution

} Any or all in combination;

e.g., Bayesian inference with memes and genes.

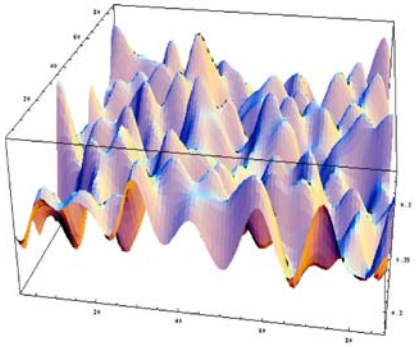
OPTIMIZATION VS. LOCAL OPTIMA

Complex systems can improve themselves through evolution,

but the solution space is too large to discover the optimum.

- No practical way of finding the optimum exists.
- Instead, you go for *viability*, i.e., something that's workable, rather than optimal.
- Might require sacrifice to make improvement
- Ultimately, what you're trying to do is maximize fitness—or survivability, score, etc.—in the face of an ill-defined future.
- This is not about accepting second best; it has to do with change and improvement in finite time.

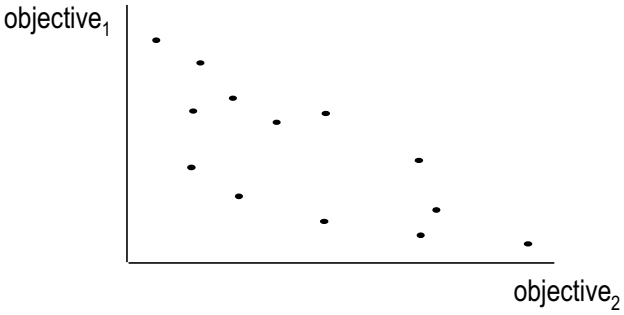
A fitness landscape: there are lots of local optima—some better than others.



OPTIMIZATION VS. LOCAL OPTIMA

With multiple objectives,

there may be more than one optimum.



Pareto optima:

No other solution exists that is better on all objectives.

Essentially, all design is *multiobjective.*

FIVE MAJOR ADAPTIVE TECHNIQUES

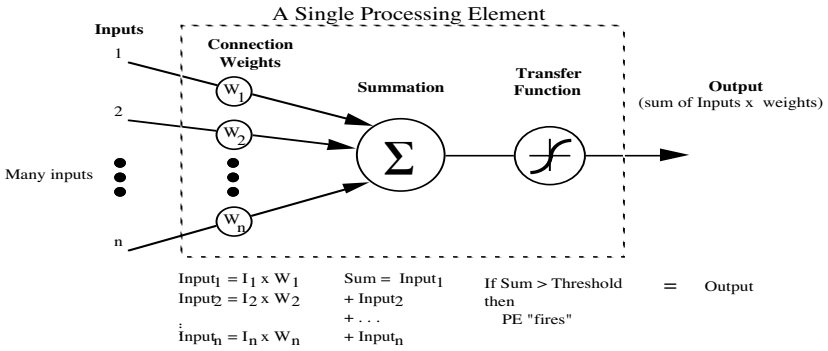
In more detail

- Neural networks
- Genetic algorithms
- Genetic programming
- Swarm intelligence
- Classifier systems

NEURAL NETWORKS

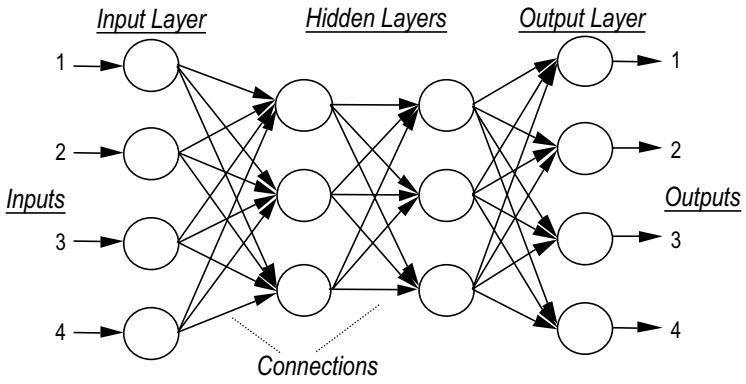
In the most common implementation, each processing element (PE)

- receives a number of simultaneous inputs,
- sums their weights, and
- determines whether the "synaptic junction" should fire or not.

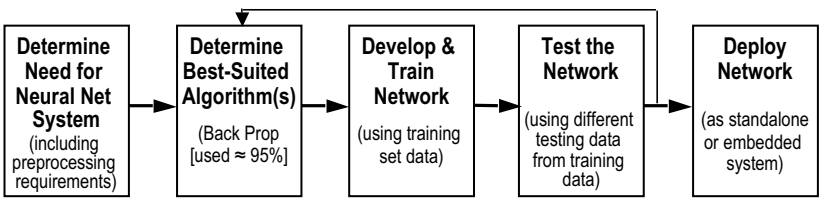


NEURAL NETWORKS

A multilayer network consisting of many interconnected PEs



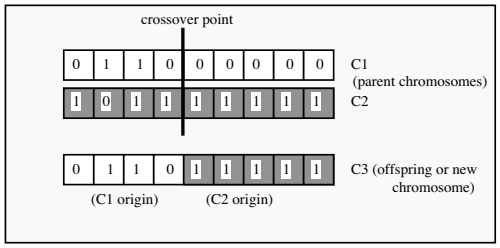
Steps in developing a neural network:



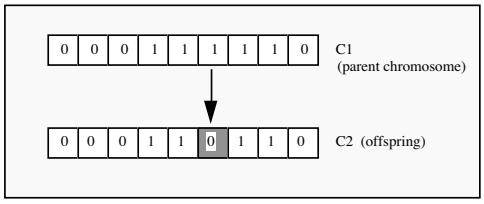
GENETIC ALGORITHMS (GAs)

Algorithm based on biogenetics and the principles of Darwinian evolution

Crossover at position 5:



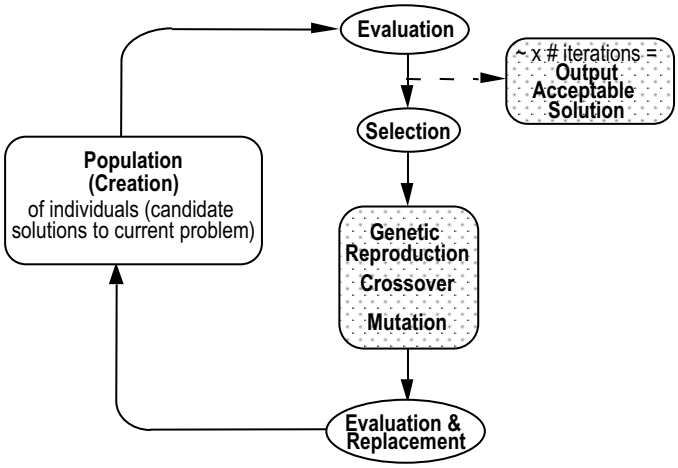
Mutation at position 6:



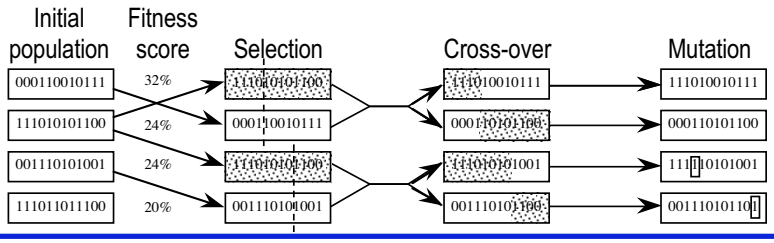
(Also referred to as *evolutionary programming.*)

GENETIC ALGORITHMS

Steps in a genetic algorithm cycle



Example:



GENETIC PROGRAMMING

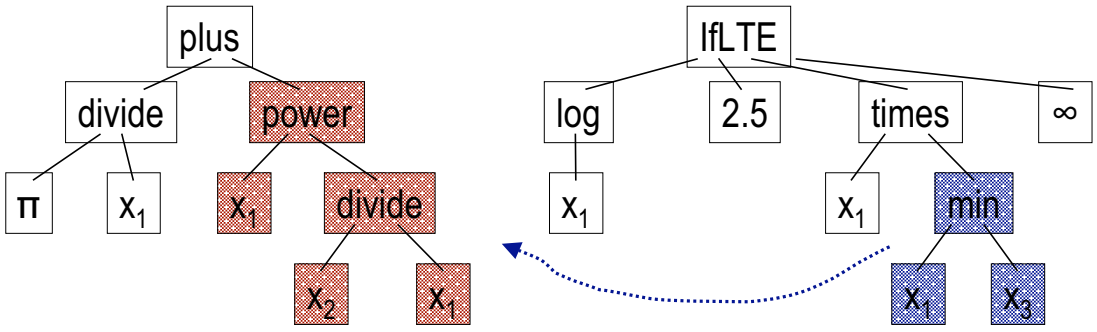
Algorithm production based on the principles of genetic algorithms

GP is a population-based evolutionary algorithm in which

- the desired solution is a function or algorithm itself, not just a set of numerical values.
- a *non-brittle* language is used to represent individual prospective solutions in the population.
- crossover, mutation, and the other search operators occur between programs, not just swapping values but exchanging arbitrary snippets of code.
- You choose population size, mutation rate, operators, representation of solutions, fitness functions, evolution constraints, ….

GENETIC PROGRAMMING

Example before crossover

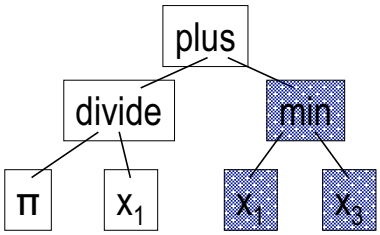


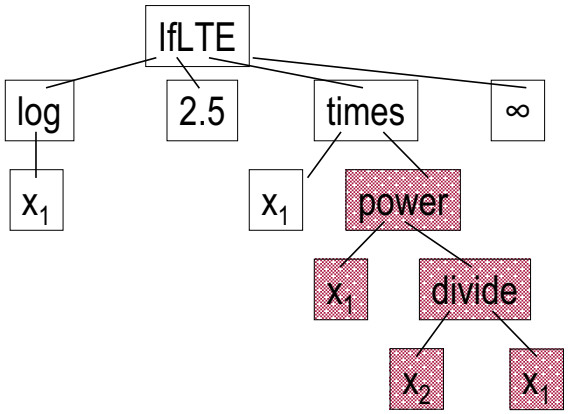
$$\frac{\pi}{x_1} + x_1^{x_2/x_1}$$

IF($\log x_1 \le 2.5$)
THEN ($x_1 \min(x_1, x_3)$)
ELSE (∞)

GENETIC PROGRAMMING

Example after crossover





$$\frac{\pi}{x_1} + \min(x_1, x_3)$$

$$\text{IF}(\log x_1 \le 2.5)$$
$$\text{THEN } (x_1 \cdot x_1^{x_2/x_1})$$
$$\text{ELSE } (\infty)$$

GENETIC PROGRAMMING

Solving quadratic equations $ax^2 + bx + c = 0$

Familiar formula: $\frac{\sqrt{(b^2 - 4ac)}}{2a}$

Given: T = { A, B, C } (3 coefficients)
F = { +, -, *, %, SQRT } (function set)

Generated code: (after 30 generations)

```
(LIST2 (% (- (% B (+ A A)) (% (SQRT (% (* (* (- (SQRT B)
   (SQRT B)) (* (+ (- C A) A) (% A B))) (* (SQRT
   (% B B)) (* (- (% (SQRT A) (SQRT (SQRT (SQRT
   A)))) (% (SQRT -1) (* C -3))) (- B B)))) (* A
   A))) (* C -3))) (- (% (% (- (SQRT (% (* (* (-
   (SQRT (% 3 -4)) (SQRT B)) (* (SQRT -1) (% A
   B))) (* (SQRT (% B B)) (* (- -2 C) (- B B))))
   (* A A))) (% (SQRT (SQRT (* (- B B) B))) (* C
   -3))) (- C A)) (SQRT -1)) (% (SQRT A) (% (SQRT
   (% (* (* (- (SQRT B) (SQRT B)) (* (SQRT -1) (%
   A B))) (* (SQRT (% B B)) (* (- -2 C) (- B
   B)))) (* A A))) (* C -3)))))
 (% (SQRT (+ (+ B -5) (+ (- (* (- (+ B -5) 1) (%
   (+ 1 B) 4)) (+ (* C A) (- (+ (% (- B B) 4) -5)
   1))) (% (+ 1 (+ 1 B)) 4)))) A)).
```

Equivalent to:

```
(% (- B) (* 2 A)) +/-
(% (SQRT (- (* 0.25 B B) (* A C))) A))
```

Koza, John R., *Genetic Programming: On the Programming of Computers by Means of Natural Selection.*, MIT Press, Cambridge, MA, 1992.

SWARM INTELLIGENCE

Technique of using the models of social insect behavior to design complex systems.

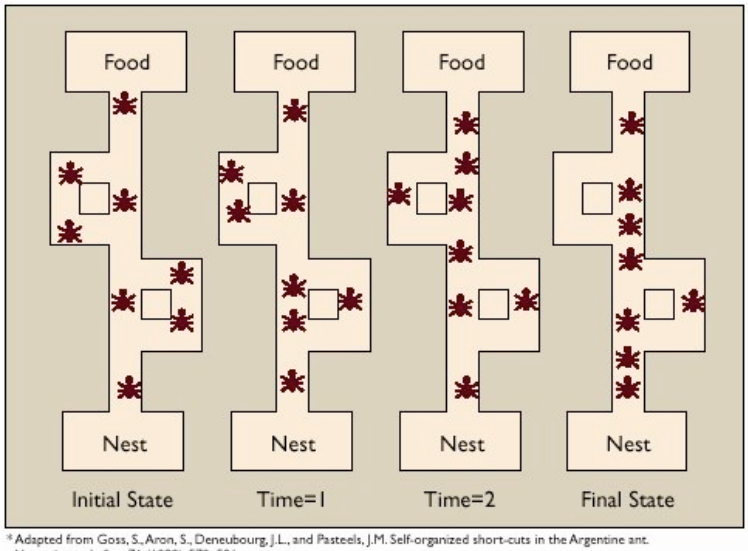
This social insect analogy for problem solving has become a hot topic in the last few years.

- It is inspired by the mechanisms that generate collective behavior in insects.
- Individual behavior modifies the environment, which in turn modifies the behavior of other individuals
- It emphasizes distribution, indirect interaction among agents,flexibility, and robustness



SWARM INTELLIGENCE METHODS

Experimental setup and drawings of the short branch selection of the *Linepithema humile* ant

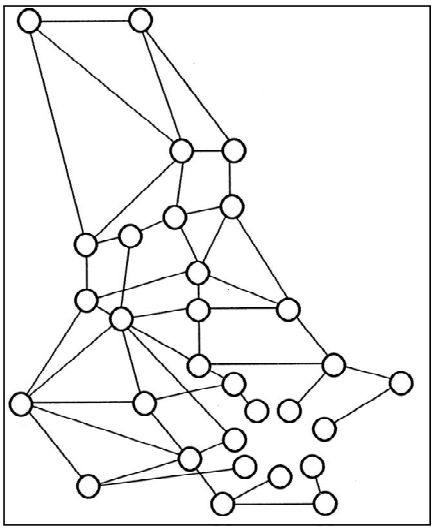


*Adapted from Goss, S., Aron, S., Deneubourg, J.L., and Pasteels, J.M. Self-organized short-cuts in the Argentine ant. *Naturwissenschaften*, 76 (1989), 579–581.

Bonabeau, E., M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, New York: Oxford University Press, 1999.

SWARM INTELLIGENCE METHODS

Network connection application

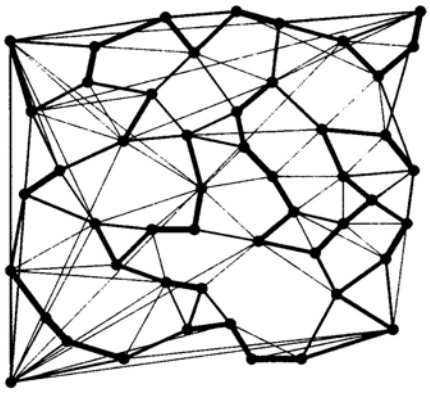


	Average call failures	Std. Dev.
Shortest path	12.53	2.04
Mobile agents	9.24	0.80
Improved mobile agents	4.41	0.85
ABC without noise	2.72	1.24
ABC with noise	2.56	1.05

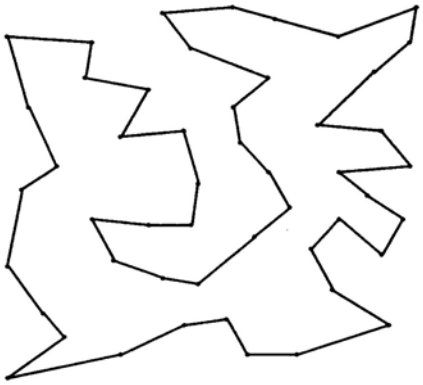
Interconnection structure of the British Telecom SDH network

SWARM INTELLIGENCE METHODS

Traveling salesman application



Pheromone concentration

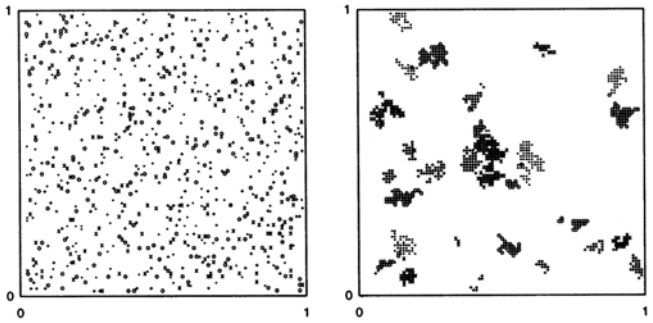


Best solution found by ants

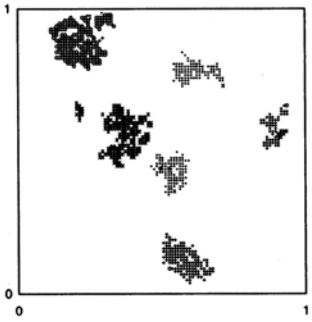
An example of a trail configuration found in a 50-city problem

SWARM INTELLIGENCE METHODS

Sorting application

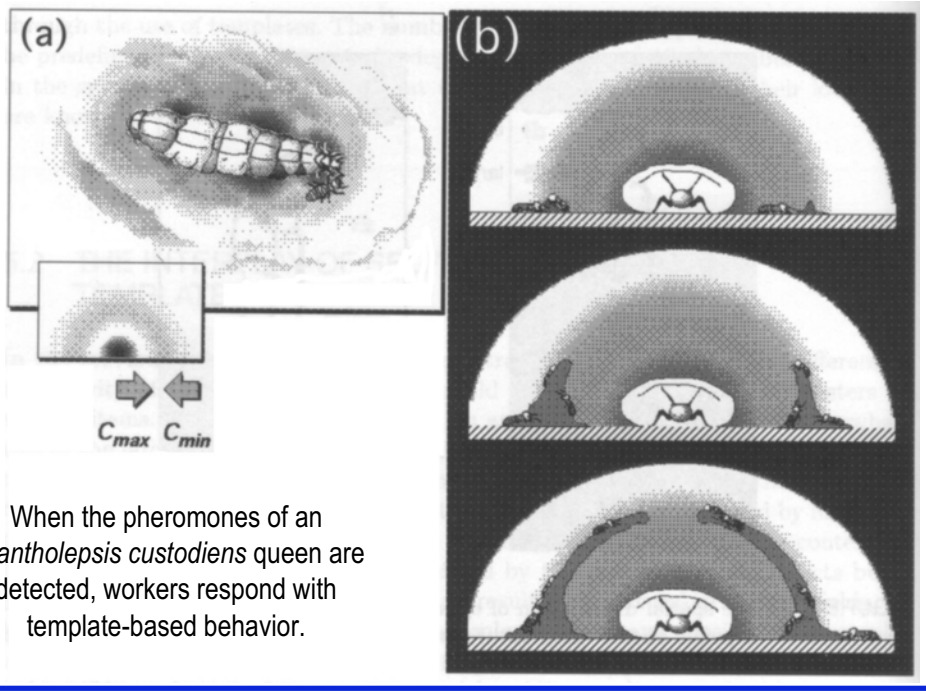


An example of sorting seven kinds of items (800 items total)



SWARM INTELLIGENCE METHODS

Pheromonal template application



When the pheromones of an *Acantholepsis custodiens* queen are detected, workers respond with template-based behavior.

SWARM INTELLIGENCE METHODS

Some of the challenges:

- Problem solving is concurrent and massively parallel.
- Poor at finding novel, counterintuitive solutions.
- Must hand-design internal ant dynamics and population size.
- Mapping from ant algorithm to emergent dynamics is poorly understood.

Problem	Genetic Algorithms	Simulated Annealing	Tabu Search	Neural Networks
Traveling Salesman	Better	Better	Same	---
Vehicle Routing	---*	Better	Worse	Better
Single Machine Tardiness	---	Better	---	---
Power Economic Dispatch	Better	---	---	---

* Not tested. Table summarized from data in [2, 4, 5, 12].

HAZARDS OF LEARNING

What is learned is not necessarily accurate.

- **Mistaking randomness for regularity** - seeing order where there is none. In humans, this is loosely identified with stereotypes—and even superstition.

- **Mistaking regularity for randomness** - seeing noise where there is order. In humans, this is loosely identified with the limitations of human senses—and even denial.

The two tend to accompany and support each other:
stereotypes invite skewed/invented regularities—
so that denial is accompanied by superstition.

ADAPTATION CONSIDERATIONS

Overview

- the basic rules and constraints
- frozen accidents
- employing phenomena that resemble
  - Darwinian evolution
  - neuron-based systems
  - parasitism
  - symbiosis
  - arms races
  - leftist Italian versus Stalinist regimes
  - mimicry
  - niche formations
  - speciation
  - sexual and asexual reproduction
  - eukaryotic versus prokaryotic cells
  - ...

ADAPTATION

Conclusion

- The problems facing agents are not well defined.
- The environment might be changing—where such changes are totally unknown.
- The solutions are not optimal.

This is what life is all about. People routinely make decisions in contexts that are not well defined—and the solutions are far from optimal.

As humans, how do we cope?

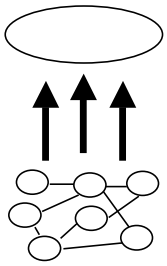
- We muddle through.
- We adapt ideas.
- We copy.
- We try what worked in the past.
- We try out new things.

Always:

- evolving
- exploring new territories
- balancing exploration with exploitation

EMERGENCE

The appearance of a coherent pattern that arises out of interactions among simpler objects—but is* more *than just their summed behavior.



- Agents organize into a larger structure whose dynamics is greater than the sum of the components' dynamics.
- 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 levels of emergence.
- Building blocks at one level can combine to form building block at a higher level.
- Collections of agents can be homogeneous or heterogeneous—emergence is a property of systems, not individuals.
- Emergence is a property of systems, not of agents.

(e.g., 1000 people with $10 million and some stock don't make a market.)

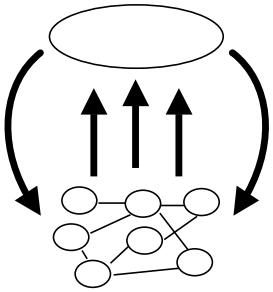
Complexity is the science of emergence.

A HIERARCHY OF INTERLOCKING MECHANISMS

Why is the world often structured in this way?

- Aggregation hierarchies are an excellent way of getting work done without being overwhelmed by the whole.
- A hierarchical, building-block structure utterly transforms a system's ability to learn, evolve, and adapt. Adaptive systems can shuffle building blocks and take giant leaps. It's more efficient than starting over from scratch.
- A monolithic structure has no components to manipulate and is, therefore, more difficult to modify naturally.
- With components, a great many complicated things can be described with relatively few building blocks.

LOCAL INTERACTION, GLOBAL DYNAMICS



- Local interaction can give rise to global dynamics—creating a coherent structure.
- The global dynamics, in turn, can influence the local interaction.
- Here, the emergent structure is linked to the local interaction,
  - influencing the boundary conditions of the local agents,
  - as a result, local agents can adjust to the presence of the global dynamics,
  - which might lead agents to change the conditions under which the agent behaves.
  - In short, a feedback loop can be formed.

COMPLEX SYSTEMS INVOLVE NONLINEAR DYNAMICS

In the human world:

- Millions of individual decisions to buy or not to buy can reinforce each other, creating a boom or recession—and can feed back to shape the very buying condition that produced it.
- Virtually everything and everybody in the world is caught up in a vast, nonlinear web of incentives, constraints, and connections.
- The slightest change in one place can cause tremors everywhere else.

If all you do is sell bananas, then you will just have a system that produces more bananas.

But, if you diversify and increase complexity above the critical point, you can expect to undergo an explosive increase in growth and innovation (economic takeoff).

Upheaval and change as well as enormous consequences can result from seemingly trivial events—yet having a deep law hidden beneath.

DESIGNING FOR EMERGENCE

Properly designed agent societies perform better than those that emerge naturally.

Organization structure is distinct from agent interaction:

- Organizational roles and responsibilities represent long-term guidelines that are subject to elaboration and revision by the agents—and offer a range of possibilities:
  - simple agents, simple tasks, simple organizations
  - workflow organizations
  - collaborative organizations
- On the other hand, the kind and nature of the interaction between agents is one of the most important factors. Without the right interaction, emergence is not possible.

Choosing the right organization is important;
choosing the right form and nature of interaction is crucial.

EMERGENCE—
USING LIFE AS AN ANALOGY

System (science)	Typical Mechanisms
Nucleus (physics)	Quarks, gluons
Atom (physics)	Protons, neutrons, electrons
Molecule (chemistry)	Bonds, active sites, mass action
Organelle (microbiology)	Enzymes, membranes, transport
Cell (biology)	Mitosis, meiosis, genetic operators
Multicellular organism (biology)	Morphogenesis, reproduction
Social group (biology)	Individuals, social relationships
Ecosystem (ecology)	Symbiosis, predation, mimicry

- Living systems are machines:
  - Instead of being designed from top down the way human engineers do,
  - living systems emerge from the bottom up from a population of much simpler systems.
- One possibility is that life isn't *just like* a computation. Life literally *is* a computation.

EMERGENCE AND PREDICTION

Look at meteorology:

- The weather never settles down.
- It never repeats itself exactly.
- It's essentially unpredictable more than a week or so in advance.
- And yet, we can comprehend and explain almost everything that we see up there. It's all just sunlight, air, water, and dirt.
- We can identify weather fronts, jet streams, and high pressure systems. We can understand their dynamics. We can understand how they interact to produce weather on a local and regional scale.
- In short, we have a real science—without full prediction. And we can do it because prediction isn't the essence of science. The essence is comprehension and explanation.

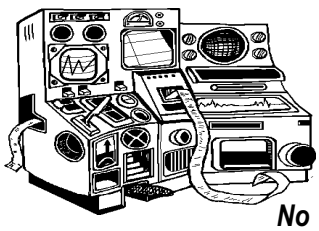
-John Holland

LIKE IT OR NOT, THE WORLD ISN'T STABLE.

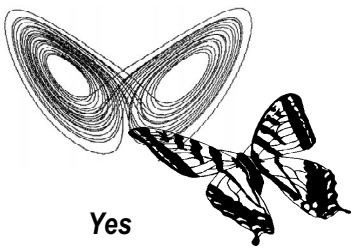
It's full of evolution, upheaval, and surprise.

- Business is not a machine, but a kind of living system with all the spontaneity and complexity of life. New products, technologies, and markets are constantly arising and old ones are dying off.
- Tiny initial differences can produce enormously different effects. Simple dynamics can produce astonishingly complex behavior.

Your company:



No



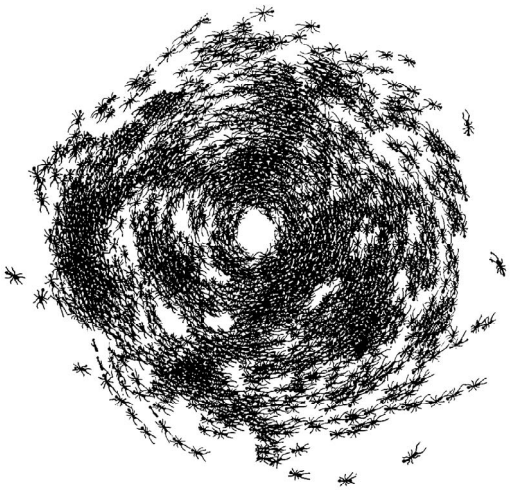
Yes

THE UNFORESEEN

Emergence happens: it can be a good thing or a bad thing.

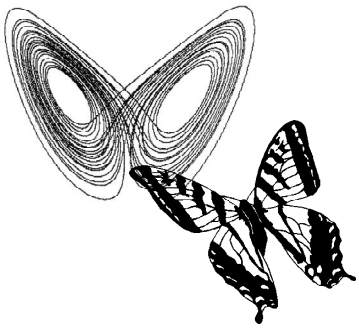
In multiagent systems, we want it to be our friend—
however, emergence is not easy to design.

A circular mill of army ants that were cut off from the colony by rain. The workers were so attracted to each other that none left the group.

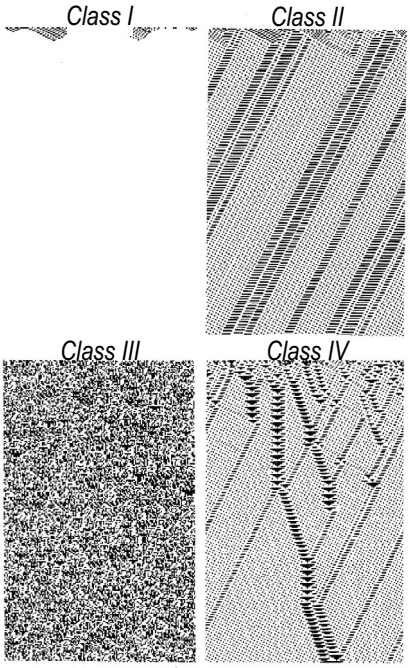


Holldobler, Bert and Edward O. Wilson, *The Ants*, Belknap Press, Cambridge, MA, 1990.

ORDER AND CHAOS



STEPHEN WOLFRAM'S CLASSIFICATION OF LONGTIME CA BEHAVIOR



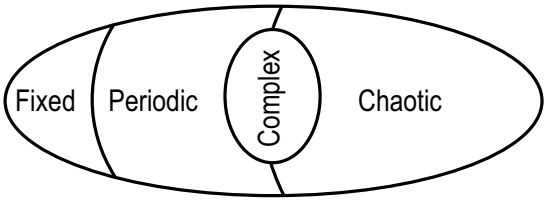
Wolfram, S., *A New Kind of Science*, Champaign, IL, Wolfram Media, 2002.

BETWEEN ORDER AND CHAOS

Fitness is maximized between order and chaos.

Fitness is low:

1) When each species is better off not changing, its evolution freezes in an evolutionary stable strategy (ESS).
2) When each species must continually change in a persistent "arms race" (Red Queen behavior).



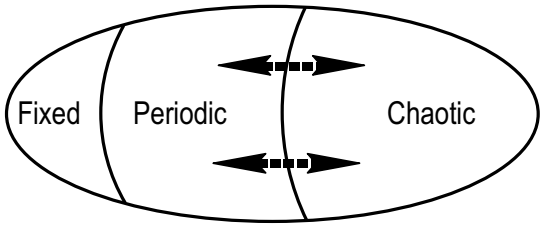
Chris Langton's "sweet spot"

The highest average fitness occurs between ordered and chaotic behavior.

"The balance point—often called *the edge of chaos*—is where the components of a system never quite lock in place and yet never dissolve into turbulence, either." — M. Mitchell Walthrop

BETWEEN ORDER AND CHAOS

Flipping back and forth between order and chaos.

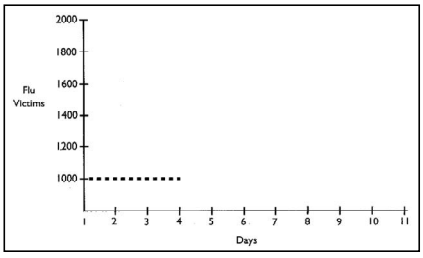


System flips across the chaotic edge where variation predominates. The system then cools, crystallizing into a new structure or behavior.

"The edge of chaos is the constantly shifting battle zone between stagnation and anarchy." — M. Mitchell Walthrop

NONLINEARITY DRIVES CHAOS

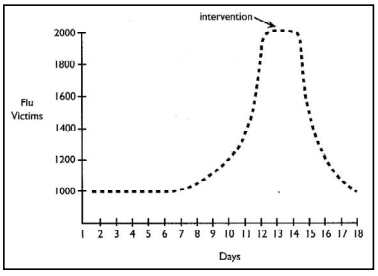
A hallmark of complex systems



linear:

1000 people meet *50* people/day

2% chance of getting 24-hour flu



nonlinear:

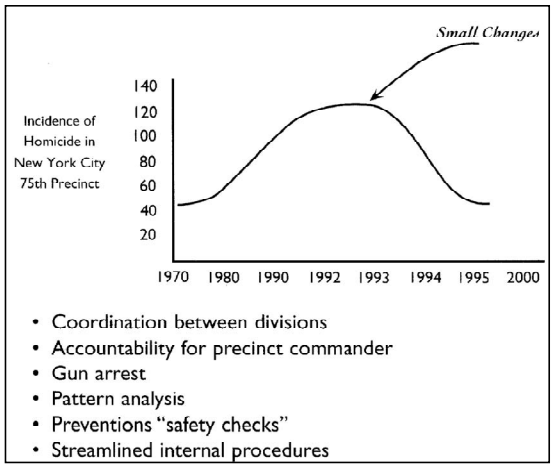
55 people/day

The job of management is to see how to intervene efficiently in non-linear systems. It must take advantage of "tipping-point" properties rather than struggling with the system as though it were linear.

Meyer, Christopher, "An Introduction to Complexity," *Embracing Complexity: Exploring the Application of Complex Adaptive Systems to Business*, Ernst & Young Center for Business Innovation, 1996, pp. 11-18.

NONLINEARITY

Tipping point or "the straw that broke the camel's back"

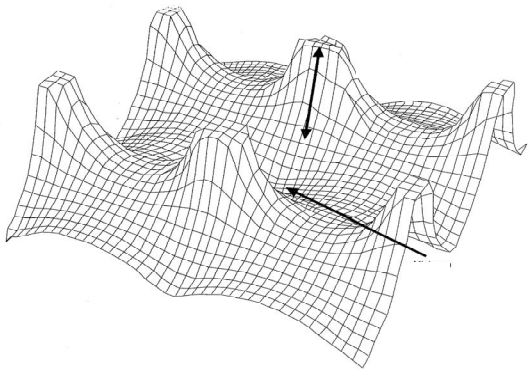


Management does not have models to identify the correct actions and reactions.
Most managerial actions fail because the system finds ways around them.

To deal with these problems, we need metaphors, theory, and tools.

The best managers are those that can find the tipping points.

COEVOLUTION LANDSCAPE

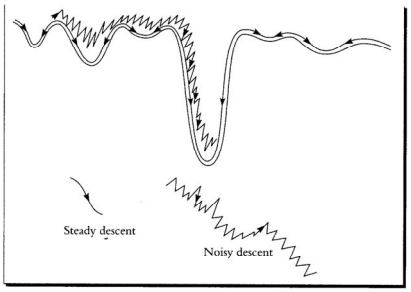


The change in an individual or its species can alter the fitness landscape for other members of the ecosystem.

This can easily become chaotic.

GREATER FITNESS BETWEEN ORDER AND DISORDER, REVISITED

A fitness landscape (with increasing fitness corresponding to greater depths)



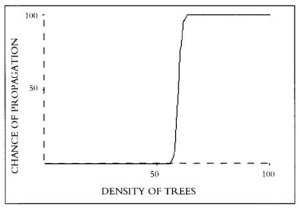
The landscape is very complicated, with numerous pits ("local maxima of fitness") of varying depths.

- If the effect of evolution were always downhill—always to improve fitness—it would get stuck at the bottom of a shallow depression and have no way to reach the deep holes of greater fitness.
- By creating some disorder, it can "jiggle" to escape shallow holes and find deeper ones nearby.
- However, too much jiggling and the whole process will cease to work. It would continue to be bounce all over and never gain greater fitness.
- A Goldilocks approach is necessary.

PHASE CHANGE

Rapid shifts from one regime to another are particularly apparent in percolation theory and collectively autocatalytic sets.

At some point, a critical density is reached when an entire forest burns.



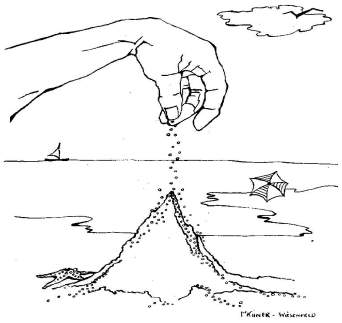
In *collectively autocatalytic sets*, the agents speed up the reaction by which they themselves are formed, two kinds of phase change can occur:

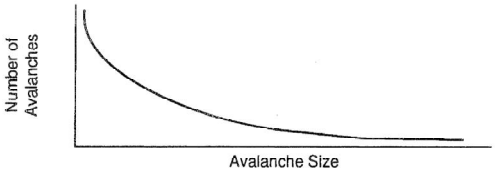
- the process goes critical and destroys itself, or
- the process eventually plateaus to a stable state.

(e.g., diseases, atom bombs, life)

SELF-ORGANIZED CRITICALITY

A pattern of self-organization in which changes occur in both time and space.





1) Avalanches can arise on all size scales; small ones are more frequent, larger less so.
2) The same initial event can trigger both small and vast avalanches

- ❑ It reaches steady state all by itself without anyone explicitly shaping it.
- ❑ The sand grains on the surface are just barely stable. (Other examples include earthquakes, stock prices, traffic jams, and extinction.)
- ❑ Power laws say nothing about a particular traffic jam, only the overall statistics. *It provides understanding, not prediction.*
- ❑ Global behavior doesn't depend on local behavior.

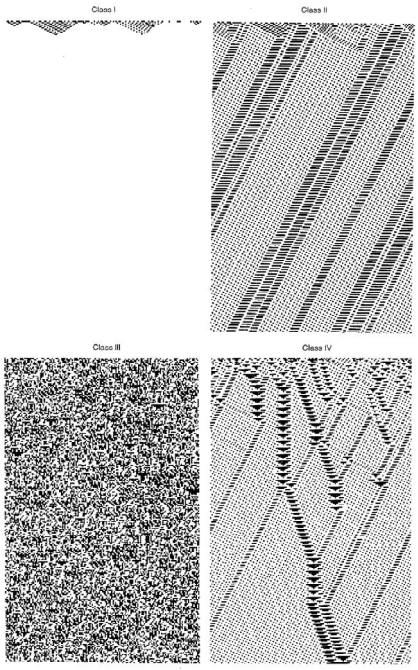
Bak, Per, *How Nature Works: The Science of Self-Organized Criticality*, Springer-Verlag, New York, 1996.

EQUILIBRIUM VERSUS COMPLEXITY

- ❑ It is meaningless to talk about a complex adaptive system (CAS) being in equilibrium; the system never gets there.
- ❑ It is always unfolding, always in transition.
- ❑ If the system ever reaches equilibrium, it isn't just stable. It's dead.
- ❑ CAS, then, are characterized by perpetual novelty.
- ❑ CAS eludes standard mathematics because CAS has multiple agents, building blocks, internal models, and perpetual novelty.

SIMULATION

Simulation is *crucial* in designing multi-agent systems which may possess emergent behavior. It will help you to anticipate how the system might behave.



DECENTRALIZATION

On December 7, 1991, Boris Yeltsin met with the leaders of the Ukraine and Belarus. After two days of secret meetings, they issued a declaration: "The Union of Soviet Socialist Republics, as a subject of international law and geopolitical reality, is ceasing its existence."

The next day, IBM chairman John Akers publicly announced the decentralization of the computer giant into more than a dozen semiautonomous business units—each with its own financial authority and board of directors.

DECENTRALIZATION

Complexity from simplicity

When constructing agent systems:

- You control the action of the parts, not the whole—or vice versa.
- You act as a designer, but the resulting pattern is *not* designed.
- Self-organizing patterns are created *without* a central designer.
- You must have many agents acting in parallel to get critical mass. A colony of ten ants will not be sufficient.
- The parts must be interacting—parallelism is not enough. Without interactions, interesting colony-level behaviors will never emerge.

Remember:

- A flock isn't a big bird.
- A traffic jam isn't just a collection of cars.

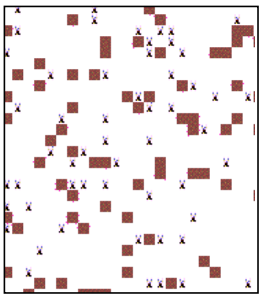
TOP-DOWN VERSUS BOTTOM-UP UNDERSTANDING

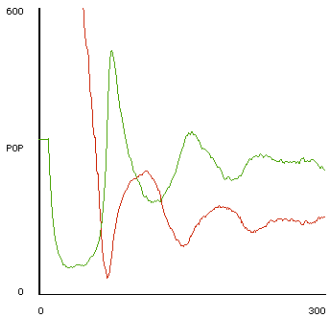
Example: Top-down, predator-prey models are based on sets of differential equations, know as the Lotka-Volterra equations:

$dn_1 / dt = n_1 (b - k_1 n_2)$, $dn_2 / dt = n_2 (k_2 n_1 - d)$, where:

n_1 = population density of prey, n_2 = population density of predator
b = birth rate of prey, d = death rate of predators, and
k_1 and k_2 are constants

Bottom-up would be creating a set of computer creatures that would interact and evolve.





Lotka-Volterra deals with aggregate quantities (population densities).

Simulation deals with the behaviors of individual creatures—from which the population dynamics emerge.

CENTRALIZED VERSUS DECENTRALIZED

Decision-makers fear that by turning control of a system over to locally autonomous agents without a central decision-making body, they will lose value that could have been captured by a more global approach.

The benefits of agent-based approaches over centralized ones are conditional, not absolute.

- In a stable environment, a centralized approach can be optimized to out-perform the initial efforts of an opportunistic distributed system.
- If the distributed system has appropriate learning capabilities, it will eventually become as efficient, although a strong enough optimized competitor may kill it before it learns enough.
- Modern market conditions are marked by rapid and unpredictable change, not stability. Therefore, we need both static and adaptable systems.

CENTRALIZATION VERSUS DECENTRALIZATION

Conclusion

- Resistance to decentralization exists. When people see a pattern, they often *assume* control is centralized.
- Centralized theories are not necessarily wrong. *Some* phenomena are described quite well this way.
- Obviously, *one solution is not right for all situations*. Neither centralized nor decentralized is *the* solution.

The ones who thrive in decentralized environments

are those who relish (not resist) unpredictability.

DESIGN FOR EMERGENCE

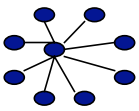


Much of this section is based on material of William Tozier

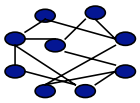
THE GOOD STUFF, SO FAR

The charm of emergence

- Emergent behavior lies at the crux of adaptability and autonomy.
- Ideally, agent-based systems should deal with their environments:
  - without constant centralized control.
  - without needing to predict every possible contingency.
  - "intelligently."



The benefits of decentralization



- Specialization and diverse behavior can occur without centralized control.
- A central controller is an information bottleneck and point of potential failure; remove it for robustness.
- Central control imposes a boundary on scaleability.
- Central components tend to attract functionality during development, inhibiting refactoring and reusability.

THE GOOD STUFF, SO FAR

The Allure of Adaptability

- Contexts and environments are dynamic on many scales
  - Fluctuations
  - Sea changes
- Robust adaptive systems cope with change automatically by
  - Reacting appropriately
  - Learning or evolving better rules

Other desirable traits…

Autonomous, interactive, mobile, proxy, sociable, proactive, intelligent, temporally continuous, credible, transparent and accountable, coordinative, cooperative, competitive, rugged, trustworthy, …

PARABLE #1
Dreams of HAL 9000

- HAL was given multiple objectives:
  1) accurate processing, without distortion or concealment.
  2) keep the mission's true goals a secret.
- HAL *adapted* to these conflicting objectives.
- Was HAL's novel solution — "Become psychotic and kill everyone" — a *bug?*

- **Moral**: Emergent behavior in designed systems is desirable and frightening.
- **Open Question**: Was there any way to *design* HAL to handle *unforeseen* conflicts like this one?

PARABLE #2

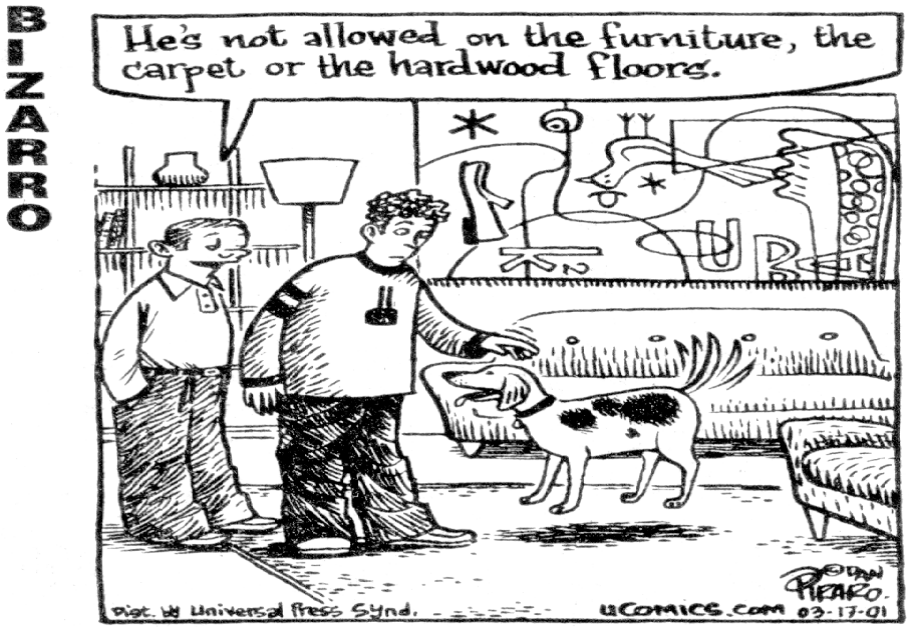
New hope for the next gas shortage

- **The Problem**: Design an autonomous adaptive pilot agent for a pilotless aircraft, using evolutionary algorithms.
  - Maximize fuel efficiency while reaching a target location and returning to base to report on what's observed.
  - Use a software-based flight simulator as a test bed, with an evolved agent as the pilot.
- **It works!**
  - A clear winner with a huge relative fitness arises.
  - The evolved pilot agent runs a surprisingly long time on just *one fueling*.
  - It has more fuel than it started with.
  - The pilot evolved to take advantage of a software-specific flaw in the flight simulator.

- **Moral**: Agents are designed to fit *models*, not the real world.
- **Open Question**: When should you start to question your models—and other received knowledge?

PARABLE #2

An agent designed to fit a model, not the real world.



PARABLE #3
Farmers' Almanac v. 2.0

- **The Problem**: Design a system that automatically discovers good farm practice for a given situation.
  - *E.g.,* "What farming methods will lead to the largest predicted harvest?"
  - Use neural nets, Bayesian belief networks, cultural algorithms, expert systems, cellular automata— anything you like from the modern arsenal

- So how about we just use an off-the-shelf evolutionary algorithm to identify the input values that result in the highest output values?
  - Best predicted harvest occurs when:
  - Plant seeds minus 5 inches deep (agro-levitator not included)
  - Apply water at 340 liters/m^2/day (wear boots)
  - Plant 890 acres on 1 May (get lots of sleep on 30 April)
  - And so on

PARABLE #3
Farmers' Almanac v. 2.0

- Using an adjusted model...
- It works!
  - Surprisingly often, the agent suggests an approach which, when implemented on a *real* farm, yields a very large harvest.
  - Farmers are happy…

- …until…
  - *Why* does a plan suggested by the agent work?
    - The system *cannot* answer the question "Why?"
  - What does it see that the farmer doesn't?
    - The agent sees and knows only the model and cannot explain using out-of-model information.
  - When will it be right and when will it be wrong?
    - …and what will happen when it *is* wrong?

PARABLE #3
Farmers' Almanac v. 2.0

- Morals:
  - *Question authority.* Good models are crucial to success of automated design.
  - Abandon hope or tune your automated design process.
  - Engineering always involves multiobjective and constrained search.

- Open question
  - If you were to design an agent so it *can* answer "why" questions, would it be intelligent?

AGENT-BASED DESIGN

- Search
- Optimization

"

Norbert Wiener, in *God and Golem, Inc.* (1964)

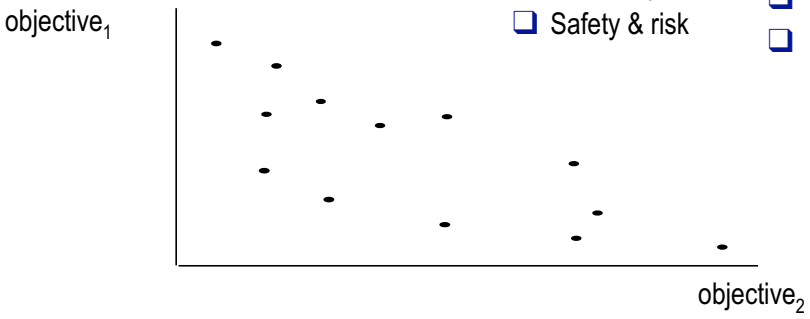
- A goal-seeking mechanism will not necessarily seek *our* goals unless we design it for that purpose,
- and in that designing we must foresee all steps of the process for which it was designed,
- instead of exercising a tentative foresight which goes up to a certain point,
- and can be continued from that point on as new difficulties arise.
- The penalties for errors of foresight, great as they are now, will be enormously increased as automatization comes into its full use.

"

IDENTIFY DESIGN GOALS

All* system engineering and design is intrinsically *multiobjective

- Functionality
- Cost
- Feasibility
- Safety & risk
- Robustness
- Extensibility & modifiability
- Versatility
- …



- Multiobjective problems often contain conflicts and trade-offs
  - Cost *vs.* safety
  - Versatility *vs.* functionality
  - Exploration *vs.* exploitation
- These trade-offs by definition demand subjective decisions—and cannot be automated. What is "best"?

ADAPTATION AXIS

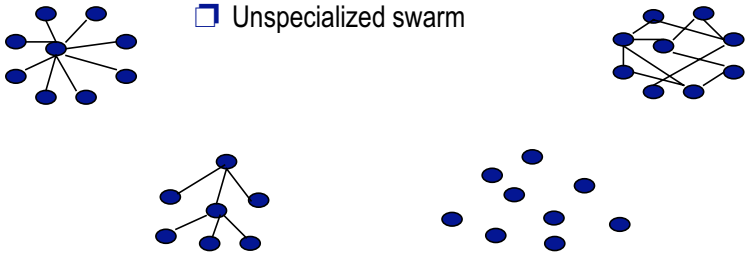
- Adapt by reaction:
  - Pre-programmed behavior, plus "physics" — inherent but emergent.
  - *E.g.,* Ant-based optimization.
  - Tune parameters offline. *Online* implies hierarchy and variable goals….
- Adapt by learning:
  - Offline (pre-training) or online (agents revise own behavior in light of performance)?
  - May over- or under-correct for fluctuating environment.
- Adapt by evolution:
  - Offline (pre-evolved) or online (evolving population of agents)?
  - Premature convergence, failure to converge, other considerations….

INTERACTION AXIS

- No interaction between agents
  - Convenient for offline adaptation: just train them all offline in a simulation with NN or GA.
  - Emergent behavior unlikely.
- Homogeneous MAS:
  - *E.g.,* ant-based systems.
  - Inter-agent dynamics easier to handle, emergent behavior possible.
  - Does the number of agents change over time?
- Heterogeneous MAS:
  - Capable of diverse interactions, complex nonlinear dynamics.
  - Hard to design by traditional approaches.

DECENTRALIZATION AXIS

- Central processing
- Hierarchy
- Small world
- Unspecialized swarm



TRADITIONAL VS REALISTIC DESIGNS

Traditional design

- Rely on comprehensive knowledge of conditions for use.
- Usually involve top-down design of all components and their interactions.
- *Assume linearity.*
- Centralized control.
- Does not *usually* require simulation to predict how it will work.

Realistic design

- Systems are open—the environment matters.
- Bottom-up design is an option.
- Frequent nonlinear interaction between components.
- Decentralized control.
- Requires simulation (because of nonlinearity).
- Cannot effectively explore space of behaviors — *emergence happens.*

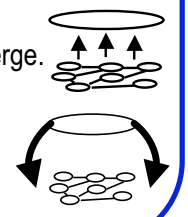
"WE'RE BUILDING AN X THAT RELIES ON EMERGENT PROPERTIES."

The charm

- Emergence is vital for discovery, learning, and design.
- Automated search can be used where traditional explicit design methods are intractable.
- Nonlinear systems can process more information more efficiently.
- Cool stuff can happen in emergent systems.

The chasm

- Emergence undermines discovery, learning, and design.
- You can design individual agents' behaviors, and the collective dynamics will emerge.
- You can design the collective's behavior, but the individuals that give rise to that behavior are difficult to design.



WORKING, BORING, OR SCARY

When is an adaptive agent *not working* — when it's acting unexpectedly, or when it's being boring?

- Well, that depends on your engineering design and testing criteria, doesn't it?
- "Traditionally," it's bad to be unexpected.

Automatic designs are scary by the old standards. The results of automatic design are frequently:

- Indecipherable.
- dependent on the platform and the implementation of the models.
- impossible to test under all conditions.

KEEP IN MIND

- Design depends on simulation and sound models. Correctly guided, it can create miraculous innovations and robust solutions.
- Adaptability of agents is governed by tendencies to exploration and exploitation, order versus chaos.
- Interactivity and guidance are *more important* when building autonomous systems.
- Never expect shrink-wrapped methods to work off the shelf. Tuning is always crucial. *If it works the first time, be afraid.*
- When a system you are working on relies on emergent behaviors, *you cannot simultaneously design both components and the whole*.
- Think about *husbanding* or *collaborating with* your creations.
- Don't expect to understand how it works — or even how it was created.

OPEN QUESTIONS

- What are "good" search methods to start with for common general classes of problems?
- Who will be liable for the actions of designed systems which are themselves autonomous and adaptive?
- How do we specify, create, and test systems which are *intended to adapt* to unforseen circumstances?
- What changes in the organization of computer systems will/musl accompany the rise of complex systems and agent-based engineering?

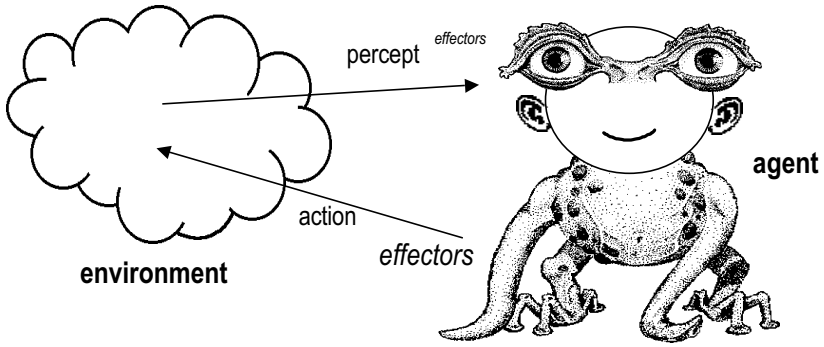
> "The world of the future will be an ever more demanding struggle against the limitations of our intelligence, not a comfortable hammock in which we can lie down to be waited upon by our robot slaves."
>
> – Norbert Wiener, *God and Golem, Inc.* (1964)

AND, IS THIS ANY WAY TO CHANGE THE EMERGENT QUALITY OF A SYSTEM?



CONCLUSION

Some ending thoughts about agents



KEEP AGENTS SMALL

- **Small in Mass** - large numbers of agents, each small in comparison with the whole system. Each termite is an almost negligible part of the entire termite hill. As a result, the behavior of the whole is stable under variations in the performance of any single member, and the collective dynamics dominate.
- **Small in Time** (Forgetful) - Naturally occurring agent systems can forget. Pheromones evaporate, and as a result obsolete paths leading to depleted food sources disappear rather than misleading members of the colony. Even the death of unsuccessful organisms in an ecosystem is an important mechanism for freeing up resources so that better adapted organisms can flourish. Furthermore, each transaction can cost and agent, leading to a decrease in energy over time.
- **Small in Scope** (Local Sensing and Action) - The participants in natural systems usually can sense only their immediate vicinity. In spite of this restriction, they can generate effects that extend far beyond their own limits, such as networks of ant paths or termite mounds. Wolves are built close to the ground and so can't see very far, but manage to coordinate a hunt with other wolves over a wide area.

Parunak, H. Van Dyke, *What Can Agents Do in Industry, and Why: An Overview of Industrially Oriented R&D at CEC*, Industrial Technology Institute, http:www.erim.org/~van/cia98.ps, 4/98.

WHEN TO USE AGENTS

Agents are not a panacea for industrial software. They are best suited for applications that are:

- **Modular -** Agents are proactive objects and share the benefits of modularity that have led to the widespread adoption of object technology. They are best suited to applications that fall into natural modules. An industrial entity is a good candidate for agent-hood if it has a well-defined set of state variables that are distinct from those of its environment and if its interfaces with that environment can be clearly identified.
- **Decentralized -** An agent is more than an object; it is a pro-active object. It does not need to be invoked externally, but autonomously monitors its own environment and takes action as it deems appropriate. This characteristic of agents makes them particularly suited for applications that can be decomposed into stand-alone processes, each capable of doing useful things without continuous direction by some other process.
- **Changeable -** Modularity permits the system to be modified one piece at a time. Decentralization minimizes the impact that changing one module has on the behavior of other modules. Agent-based architectures permit reuse of much existing code and self-configuration of large portions of the system, reducing both the cost and the time needed to bring up a new factory.

Parunak, H. Van Dyke, *What Can Agents Do in Industry and Why: An Overview of Industrially Oriented R&D at CEC*, Industrial Technology Institute, http:www.erim.org/~van/cia98.ps, 4/98.

WHEN TO USE AGENTS

- **Ill-Structured -** That is, not all of the necessary structural information is available when the system is designed. Agents can discover this structure as the system operates, rather than requiring it to be designed into the system initially.

  A system that is designed to a specific domain structure will require modification if that structure changes. Agent technology permits the analyst to design a system to the classes that generate a given domain structure rather than to that structure itself. Thus, the useful life of the resulting system is extended and the cost of maintenance and reconfiguration is reduced.
- **Complex -** Consider 100 agents, each with 10 behaviors, each behavior requiring 20 lines of code. The total amount of software that has to be produced to instantiate this system is 20,000 lines of code, a modest undertaking. But the total number of behaviors in the repertoire of the resulting system is on the order of 10 for the first agent, times 10 for the second, times 10 for the third, and so forth, or 10^{100}, an overwhelmingly large number. By adopting an agent approach at the outset, systems engineers can provide a much more robust and adaptable solution that will grow naturally to meet business needs.

HEAD CHANGES

- **Central versus self organization -** who else is doing it, if it is not doing it itself?
- **Paradigm changes -** you must be willing to change your set of concepts.
- **No attainable optimum -** you can do hill climbing, but there is no guarantee you will end up on the tallest mountain.
- **No duality -** we are all part of an interlocking network.
- **Living on the edge of chaos -** not too much and not too little.
- **Integrative thinking -** No complex, nonlinear system can be adequately described by dividing it up into subsystems.
- **Large changes are OK-** "Humanity is moved forward *not* by people who stop every little while to gauge the ultimate success or failure of their ventures, *but* by those who think deeply about what is right and then put all their energy into doing it."
- **Whole systems -** not to a single individual or gene. The agent-based approach points us away from isolated units and toward interactions between individuals and their environment.
- **Viability -** something that's workable, rather than what's optimal.
- **In agent-based systems, everything changes, nothing is the same.** Yet there are patterns that are recognizable. A lot of time is often involved in coming up with appropriate metaphors.

WHERE IS AGENT TECHNOLOGY NOW?

- It is still an active research area.
- Isolated pioneer products are emerging.
- The full set of technologies is not available.
- The technologies are not integrated to one another.
- There is no consensus on how to support agents at the operating system level.
- Despite the hype, agent technology is not widely used—nor has it been widely accepted as an inevitable trend.
- There are early adopters who can demonstrate the value of agent technology.

AGENT STANDARDIZATION

- **AgentLink** is Europe's ESPRIT-funded Network of Excellence for agent-based computing. It is a coordinating organization for research and development activities in the area of agent-based computer systems aimed at raising the profile, quality, and industrial relevance of agent systems in Europe. (www.agentlink.org)
- **AgentCities**
  A global, collaborative effort to construct an open network of on-line systems hosting diverse agent based services. (www.agentcities.com)
- **Methodology for Engineering Systems of Software AGEnts (MESSAGE)** MESSAGE is a EURESCOM project developing an agent-oriented methodology that encompasses requirements analysis and design, and which is appropriate for use in mainstream software engineering departments. (http://www.eurescom.de/public/projects/P900-series/p907)
- **Java Agent Services (Sun JSR 000087)** is a specification that defines a set of objects and service interfaces to support the deployment and operation of autonomous communicative agents. Expert Group Membership: Fujitsu, Hewlett-Packard, IBM, Sun Microsystems (www.java.sun.com/aboutJava/communityprocess/jsr/jsr_087_jas.html)
- **Holonic Manufacturing Systems (HMS)** consortium's (hms.ifw.uni-hannover.de) task is to employ the analogies of social organizations and living organisms (originally inspired by Arthur Koestler's "holons") to attain for manufacturing systems benefits.

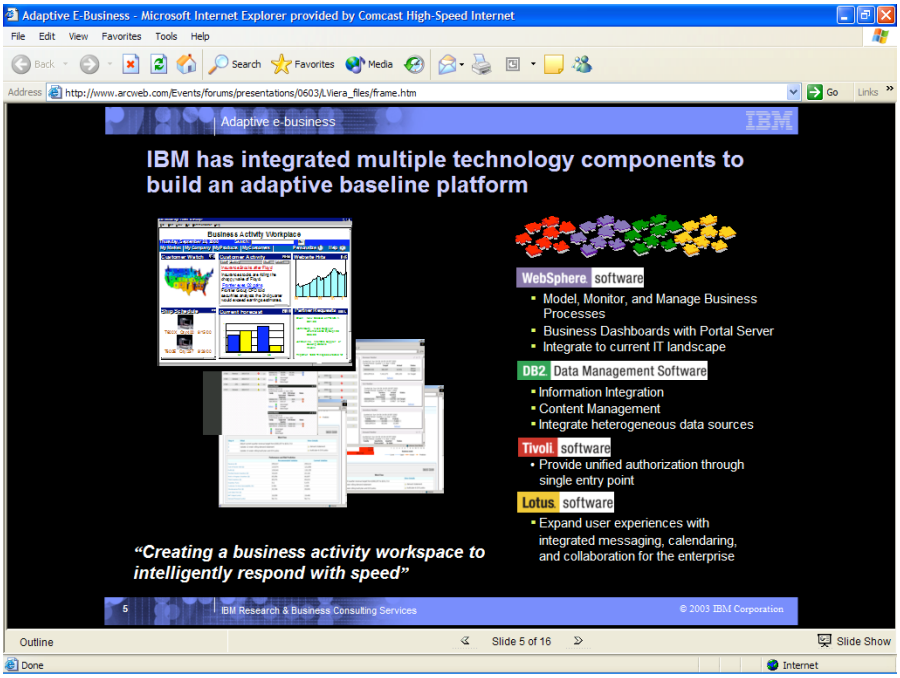
AGENT STANDARDIZATION

- **OMG Agents Special Interest Group** recommends standards for agent technology where appropriate— particularly the OMG's Object Management Architecture (OMA). (www.omg.org)
- **FIPA (Federated Intelligent Physical Agents)** has been working to develop and promote standardization in the area of agent interoperability since 1996. It has an on-going work program, meeting around the globe on a quarterly basis, with over 50 member organizations. (www.fipa.org)
- **US DARPA (Defense Advanced Research Projects Agency)** has several research programs that address aspects of agent technology. These include:
  - Control of Agent-based System (CoABS) (coabs.globalinfotek.com/)
  - Advanced Logistics Project
  - DARPA Agent Markup Language (www.daml.org)
- **KQML (Knowledge Query and Manipulation Language)** is a language and protocol for exchanging information and knowledge. It is part of a larger effort, the ARPA Knowledge Sharing Effort. (www.cs.umbc.edu/kqml/)

WHO IS WORKING WITH AGENT TECHNOLOGY NOW?

- In Europe, nearly 100 projects are now funded. (http://www.agentlink.org/resources/agentprojects-db.php)
- More than 100 software tools are now listed, academic and commercial. (http://www.agentlink.org/resources/agent-software.php)
- Some particularly active US companies:
  - Altarum (www.altarum.org)
  - IBM
  - Intelligent Automation (www.i-a-i.com)
  - Sun
  - WebV2 (www.webv2.com)

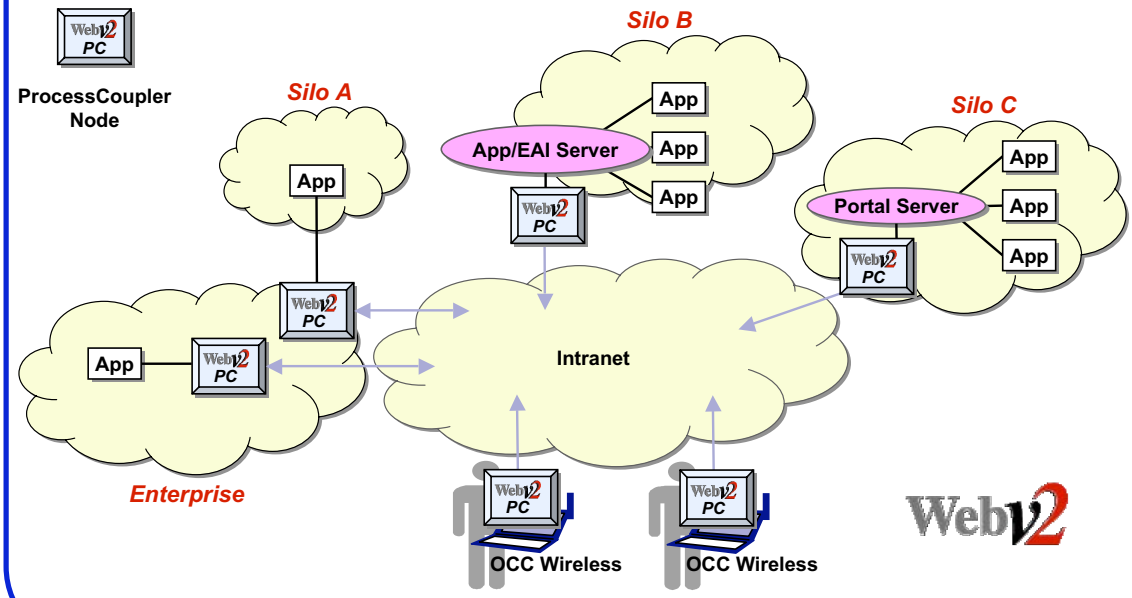
IBM



http://www.arcweb.com/Events/forums/presentations/0603/LViera_files/frame.htm

WebV2

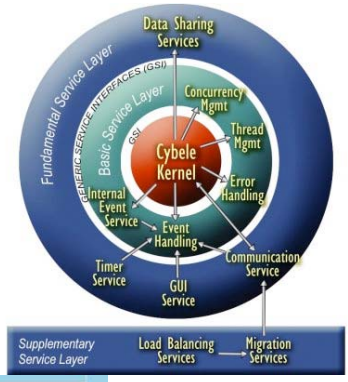
ProcessCoupler deployment within an enterprise and across multiple enterprises

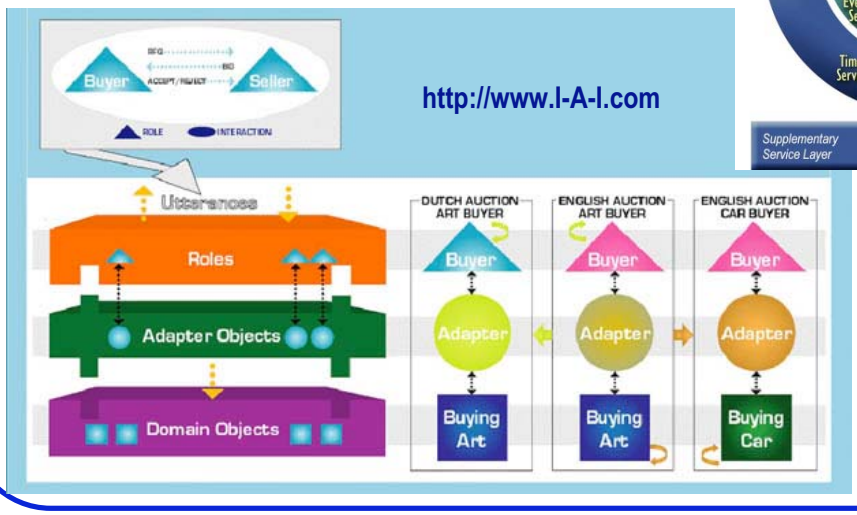


http://www.webv2.com

INTELLIGENT AUTOMATION, INC.

Cybele and DIVA use an extension of the UML model language to generate large multiagent systems.

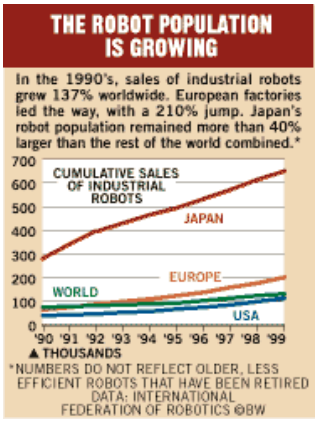




http://www.I-A-I.com

AND WHAT ABOUT ROBOT AGENTS?





AND WHAT ABOUT ROBOT AGENTS?



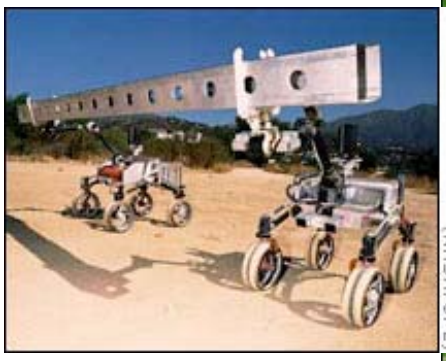






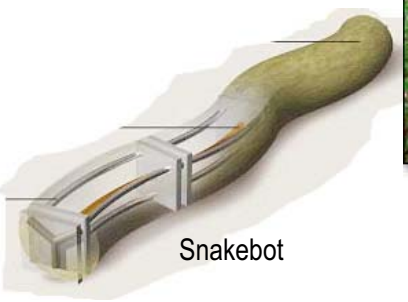
AND WHAT ABOUT ROBOT AGENTS?

BEAM (Biology, Electronics, Aesthetics, and Mechanics)
Robotics
http://www.nis.lanl.gov/projects/robot//









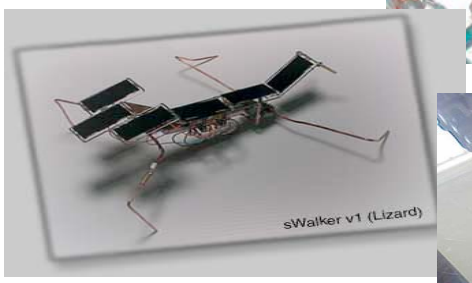
Snakebot



AND WHAT ABOUT ROBOT AGENTS?

Mark Tilden's bots









http://www.wowwee.com/biobugs/biointerface.html# and http://www.beam-online.com/Robots/vwalker1/

AGENTS WILL CHANGE:

- the entire software industry—not to mention business, in general
- the way we use computers—as well as people and machines
- the way we use networks
- the way we analyze and design systems
- the way we use automated tools
- the way we re-engineer corporations
- the job of all software developers

Potential benefits of agent technology, in general:

- Improved efficiency
- Improved effectivity
- Increased transparency and optimizations

SUMMARY

- **What is an agent?**
  - An autonomous entity that can adapt and interact with its environment.
  - It can provide massive distributed processing that is autonomous, adaptive, and interactive.
  - Can (and should) be used with objects and other system entities.
- **Why do I care about agents?**
  - Provide an *alternative* to centralized control. (It does not replace central control; it gives you one more option than you have now.)
  - Enable a new class of applications that was not previously possible; i.e., it can handle many problems that could not be handled before.
  - Security becomes more of a problem, not less.
  - Once unleashed can be difficult to fully control—and that is both the bad news *and* the good news.

TUTORIAL ROADMAP

- What is an agent?
- Agent anatomy
- Agent communication
- What gets agented?
- Adaptation
- Emergence
- Order and Chaos
- Decentralization versus centralization
- Designing for emergence
- Conclusion

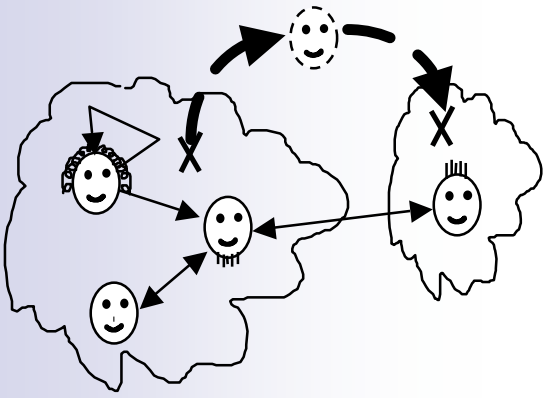
… and punctuated by lots of examples and animations

CONCLUSION

- The agent-based approach is an evolution, not a revolution.
- Start simply and build—but start now.
- Using agents is not a matter of if, but when.

A Tutorial for NODE 2003

Beyond Objects: Unleashing the Power of Adaptive Agents



Presenter

James Odell

James Odell Associates
3646 W. Huron River Drive
Ann Arbor, Michigan 48103-9489 USA
telephone: +1 (734) 994-0833
email: email@jamesodell.com
website: www.jamesodell.com