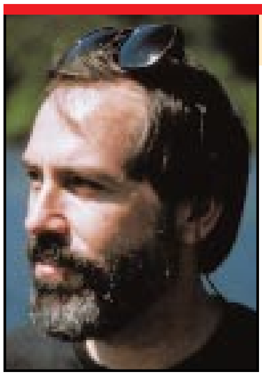
COMPLEX ADAPTIVE SYSTEMS



James Odell
jodell@compuserve.com

Designing Agents: Using Life as a Metaphor

Metaphors enable us to use one concept in place of another to suggest a likeness. They are particularly useful for introducing new ideas to an existing area—thereby expanding and enriching it. Using life as a metaphor is already quite common when discussing agents because of they can interact in an autonomous manner. As such, we can think of agents as being subjected to stress from environmental pressures, resource shortages, and restriction of growth. We can imagine them with the ability to evolve their behavior by developing ways of coping with such stresses. Some agents will survive and succeed by growing, increasing their ability to command resources, and reproducing; those that fail will shrink, be replaced, swallowed, absorbed, or die.

In short, we can use life as a metaphor to develop agent-based systems, whether the agents are software, hardware, equipment, corporate entities—or even people.

Agents and Scalability

One of the most difficult challenges for automated systems is scalability. Here, life-as-a-metaphor brings with it many useful concepts, including some excellent examples on how to scale up. In the physical systems leading up to life, for example, subatomic particles form atoms, and atoms cluster to become molecules in solid, liquid, and gaseous form. Continuing up this hierarchy, molecules can be organized to form organelles and cells, cells can aggregate to form organisms, and so on (Table 1).[1] In other words, living systems and their components emerge in a hierarchy of interlocking mechanisms.

In a very general sense, complex adaptive systems are large and intricate systems involving active, autonomous agents. Such a hierarchy is a necessary—and some say, a natural—occurrence for complex adaptive systems. Without such a hierarchy, scalability would not be possible. Life would not be possible. Without a more complex formation, 1050 subatomic particles floating around our universe would just be 1050 subatomic particles floating around. Individually, they are still subatomic particles and no more. To create something more complex, there must be a way to produce a new structure that is more than just the sum of its particles.

However, these new aggregate formations can not become too large. Such structures can easily become unstable and collapse under their own weight. For example, increasing the size of a molecule a trillion times to produce something substantive is as impractical as building sand castles a mile high.

So, sheer numbers without organization are unmanageable and bigger is not necessarily better. Yet, life's hierarchy of interlocking mechanisms provides the right mortar between the bricks to construct viable structures. Furthermore, the new structures can become building blocks for even larger structures—where each level of the hierarchy is very different than the one before and the one after it. For example, hydrogen and oxygen have very different properties than a water molecule which comprises them. A cell has a different structure and behavior than the molecules and organelles that comprise it, and so on. Why can't the same approach work for automated agents?

Table 1. A Hierarchy of Interlocking Machanisms

System (science)	Typical Mechanisms
Nucleus (physics)	Quarks, gluons
Atom (physics)	Protons, neutrons, electrons
Gases and fluids (physics)	Flows, circulation, turbulence
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

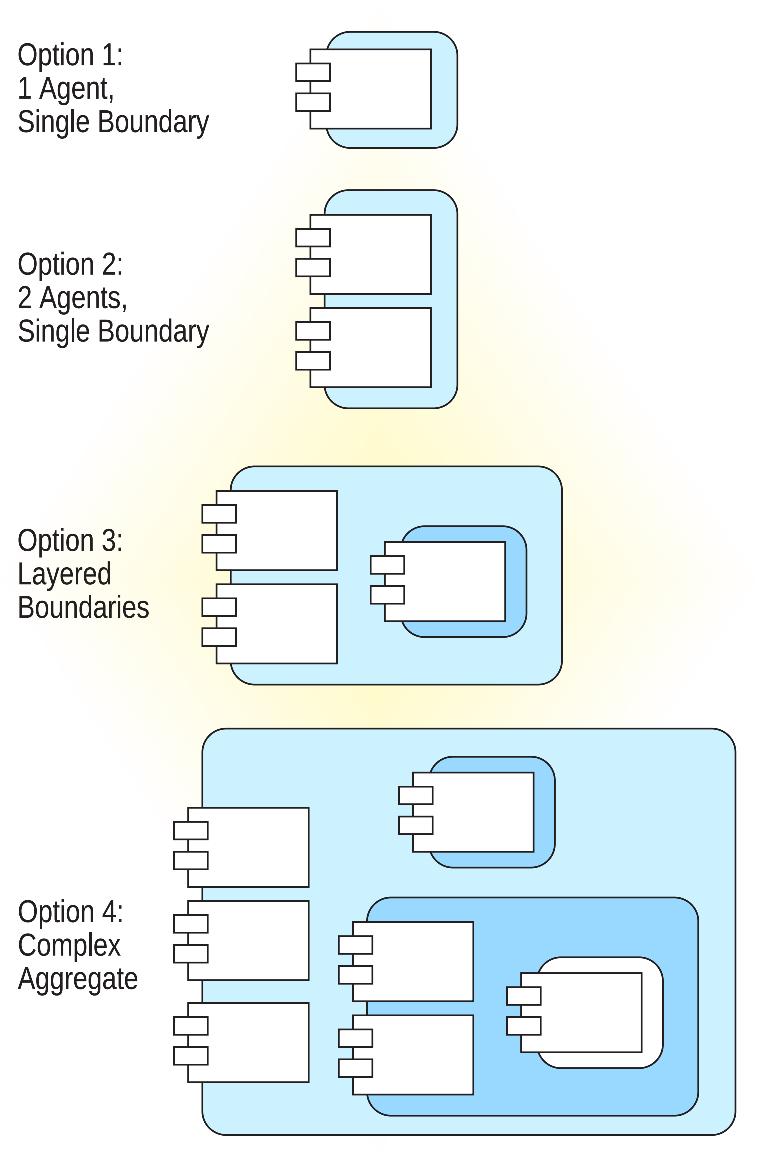
Multiagents

Agents can be aggregated to form variable structures called multiagents. These aggregates can be colonial in nature (such as sponges and coral reefs) or metazoan (that is, multicellular animals). Agents that adhere to one another can behave in a unified manner and still maintain their own autonomy. Automated agents might choose to aggregate for various reasons, such as protection, resources, or improvement. In other words, the agents might decide that they are better off together than apart. Furthermore, the agents may adapt to serve the aggregation as a whole.

James Odell is a consultant, educator, and author. He works with James Martin & Co. and IntelliCorp.

Figure 1 depicts several aggregation options.[2] The first option, of course, is that there is no aggregation. The second option is that two or more agents can aggregate as a single unit. The containment boundary indicates that the entire construct can be treated as an agent in its own right. In option 1, the agent and the boundary are the same. In option 2, the boundary can be treated as an agent separate from the two agents it contains. The term boundary is chosen because it also provides an interface barrier much like that of a cell membrane. Agents on the membrane (as in option 2) are still accessible from the outside as individuals while benefiting from the proximity to chosen neighbors. In contrast, agents that are contained completely within the boundary, as illustrated in option 3, are encapsulated. Only those agents on or immediately within a boundary may communicate with encapsulated agents. However, the boundary itself may have specialized rules that permit the passage of agents through the boundary—either into or out of the agent. (In fact, the membrane could be constructed to allow implicitly the passage of "substrate" that it does not see or care about. Granted this "breaks" encapsulation, but that's life.) In a complex aggregate configuration (option 4), the nesting continues.

Figure 1. Some Forms of Agent Aggregation



As we saw in Table 1, this is a common phenomena: molecules emerge as aggregations of atoms, cells emerge as aggregates of molecules and organelles, and so on. For the world in general, it is common to see building blocks at one level combining to form building blocks at a higher level. At each level, new structures form and engage in new emergent behaviors. The study of complex systems, then, is also the study of emergence.

In object orientation, this is consistent with how components are handled. Each component has its own interface. Those classes and components contained by the component are encapsulated, that is, they are not directly accessible by the external classes and components. While there is no equivalent OO support for option 2, it just means that the component interface includes the interfaces of those classes on the component boundary. In business, autonomous agents can be grouped into aggregates. At the Industrial Technology Institute, for example, Van Paranak (www.iti.org/~van) suggests that sensor and actuator agents can be grouped into a machine, several machine agents can form a work station, and work stations can aggregate into a manufacturing cell. In other words, grow by chunking instead of starting with large, complex agents. Favor smaller specialized agents over more general ones. Small individual agents are easier to construct and understand than monolithic ones, and if they fail, their impact will be minimal.

Agents and Distributed Control

Distributed control is also a fundamental mechanism of life-forms. It provides an alternative to having a single elaborate control center directing every single task by having multiple structures that specialize in their own subtasks. Furthermore, each of these structures may consist of many substructures, offering a finer granularity of specialized control.

Distributed control of any complex system, then, has many advantages. This is especially true when system components are widely dispersed, as in a communication, transportation, or banking network. Completely centralized systems require two-way communications links with all components. In any situation subject to rapid change, a completely centralized control requires high bandwidth communication links, a powerful central computer, and an elaborate operations control center. However, all of these are subject to disruption at any time by system bugs, natural disasters, espionage, or stress-related events. In situations where fast response and rapid recovery is important, distribution of control is usually preferable. Here, as much control as is practical is delegated to the local level. In this way, when a failure occurs, each component can act as an independent agent. If these agents have adaptive capabilities, they can organize themselves and make efficient use of whatever resources remain.

An even richer model involves autonomous adaptive agents that partly cooperate and partly compete with each other in their local operations. Industries that have multiple ownership and management often operate with a minimum of regulation—while depending on the goodwill and cooperation of all parties. Dr. Martin Wildberger of EPRI in Palo Alto (mwildber@epri.com) has demonstrated the feasibility of these kinds of agents in computer simulations of a deregulated power industry.

Agent Sensors and Effectors

Life-forms can sense their environment via an assortment of stimuli and can also effect changes in their environment. Agents, too, can have sensor and effector mechanisms. There are several ways agents can "sense."

Events

Events are changes in the environment that might be noteworthy to an agent. They can be sent directly to the agent on a broadcast basis or a subscription basis or be directly observable by the agent in its network interaction. A broadcast event can be sent to one or more agents without the receiving agent's request. In contrast, an agent may define the kinds of events that it wishes to know about by notifying its external world via a subscription. Instead of being reactive to its environment, an agent can be proactive. A proactive agent can selectively scan its environment for specific kinds of events whenever it wishes. Some agents employ both approaches.

Direct communication

Life-forms need to "communicate" with their world to determine if other nearby entities are edible or are potential partners for procreation, protection, or symbiosis. For an automated agent this might mean locating parts to ship (Order), finding the best possible price on a contract for electricity (Energy Purchase Contract), or web spiders finding the right kind of requested information. Communication between specific agents may be direct or indirect. Direct communication can be one or two way. An agent can request information from another agent, or it can provide information to that agent with or without any expectation of acknowledgment or response. (In my next column, I will discuss some mechanisms for direct communication.)

Indirect communication

Life-forms can gain a good deal of information about their world without directly communicating with another agent. For example, ants leave a pheromone trail that guides other ants to find food. British Telecom uses a similar technique to optimize call routing. Previously completed calls leave "trail markers" that indicate paths for successfully reaching certain calling destinations. Sensing, then, can play a major role as it does for life-forms—and such senses are not limited to merely the five human ones. A good example is the recent discovery of a kind of auditory "sight" via sonar. Here, dolphins not only do echolocation (ping-reflect) and passive listening, but they probably use a form of signal processing akin to triangulation in vision. In doing so, they can process the reflection of ambient noise in the surrounding waters. This is not really broadcast (where the agent emits a signal) and it is not really subscription (where the observer probes the object). It is more like continually updating their knowledge of the immediate environment. The difference in which an agent senses the world (like the difference between ambient response and broadcast) can make a substantial difference in the way an agent behaves.

Finite communication

Organisms cannot know everything about their vast world, but they can scan their neighborhood for friend or foe. Similarly, it is probably not feasible (or useful) for an electric company to know about all possible sources of electricity everywhere. Instead, its Order agents would primarily be interested in energy sources within its own neighborhood. Each agent, then, must have the ability to "sense" particular kinds of agents within a given range. For an autonomous agent to sense things in its environment, it is often useful to treat the environment as an agent in its own right. An agent can then communicate directly with the environment to learn more about its state.

Effectors

Effectors are operations that are invoked by agents. Sometimes effectors are invoked in response to certain events and under certain conditions. (The two are often linked in the form: `WHEN event, IF condition, THEN operation`. This form will be discussed more in a future column.) For example, an ant might have the following rule: `WHEN food is found, IF I am not already carrying food, THEN pick up as much food as I can carry and take it back to the nest`. Or even, `WHEN I am carrying food, THEN leave a pheromone trail`.

Effectors can be proactive, as well. For example, an Accepted Order knows that it must be filled and shipped. Such a goal-directed agent would try to ensure that order-filling and shipping effectors are invoked appropriately. (In the next column, I will discuss some of the basic forms of effector mechanisms.)

Conclusion

Using living systems as a metaphor suggests many mechanisms for designing systems of autonomous agents. This column explored just a few of those mechanisms already being applied in complex adaptive systems. Other notions that will be explored in the next column include getting and giving resources, transforming resources, interaction, rejection, pursuit, enablement, protection, adhesion, reproduction, evolution, and even death.

From the Autonomous Agents at Rock Island Arsenal (AARIA) website (www.aaria.us.edu) are some compelling reasons to use agents to develop software:

- Agents are consistent with the object-oriented paradigm. The efficiencies of programming with agents begins with the efficiencies of the object-oriented paradigm.
- A multiagent heterarchy matches the vision many have for the future of Internet computing. The idea of intelligent entities communicating and coordinating with each other over wide area networks is a common concept in the Internet community.
- Multiagent systems can be designed to be self-configuring. Agents can be added and subtracted from the system while it is running no external intervention required.
- Self-configuration and decentralization provide fault tolerance. A system with autonomously functioning components will not collapse when one or more of the components fail or malfunction.
- Multiagent architectures are inherently scaleable and modular. It is substantially less expensive from a hardware perspective to use a large number of inexpensive processors than a single processor having equivalent total processing capabilities.

References

1) Holland, John H., Emergence: From Chaos to Order, Addison-Wesley, Reading, MA, 1998.

2) Holland, John H., Hidden Order: How Adaptation Builds Complexity, Addison-Wesley, Reading, MA, 1995.

Acknowledgments

The author would like to thank Roger Burkhart (Deere & Co.) and Glen Ropella (Swarm Corp.) for their help and guidance in producing this column.