

Agents and Complex Systems

James Odell, Independent Consultant

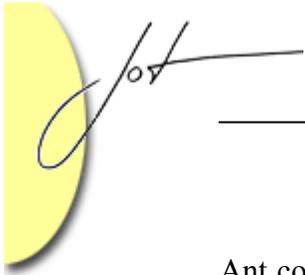
Traditional objects can be thought of as passive, because they wait for a message before performing an operation. Then, once invoked, they execute their method and go back to “sleep” until the next message. A trend in many systems now is to design objects that react to events in their environment, as well as be proactive. In UML 2.0, these are known as *active objects*; in the agent community, they are known as *agents*. Whether they are called active objects or agents, this new direction is going to radically change how we design systems. The biggest challenge, however, that we face is the degree of complexity that we are about to unleash. Imagine setting free a million proactive entities to run a supply chain. We are no longer choreographing their every movement as we would with traditional agents; instead, *they* decide when and how to execute their methods. This is both liberating - and scary. We can create complex systems, but we will not always know how to control them. In complex systems: designing the parts is not the same as designing the whole.

1 COMPLEXITY

Complexity is all around us. It is part of our life; it is the nature of life. Complexity is caused by the collective behavior of many basic interacting agents. Such agents can produce everyday phenomenon such as ant colonies, traffic jams, stock markets, forest ecosystems, and supply chain systems. Complex systems, however, do not have to be complicated. For example, the ant colony simulation of StarLogo (<http://www.media.mit.edu/starlogo/getstar-new-other.html>) illustrated in Fig. 1 is not complicated. Here, each ant has three simple rules:

1. Wander randomly.
2. If food is found, take a piece back to the colony and leave a trail of pheromones that evaporates over time; then go back to rule 1.
3. If a pheromone trail is found, follow it to the food and then go to rule 2.

In Fig. 1(a), the ants are just emerging from the ant hill to begin their random walk. Eventually, an ant discovers a food source and returns some to the colony, leaving a trail of evaporative pheromones (Fig. 1(b)). Figure 1(c) shows the ant colony well underway in retrieving food. Lastly, Fig. 1(d) depicts two very depleted food sources and one that is exhausted altogether.



Ant colonies, then, are not complicated. A system is complicated if it can be completely described in terms of its huge number of components. On the other hand, a system is complex if the system cannot be fully understood by analyzing its components; here, the interaction among the components also must be considered. [Brown 1889],

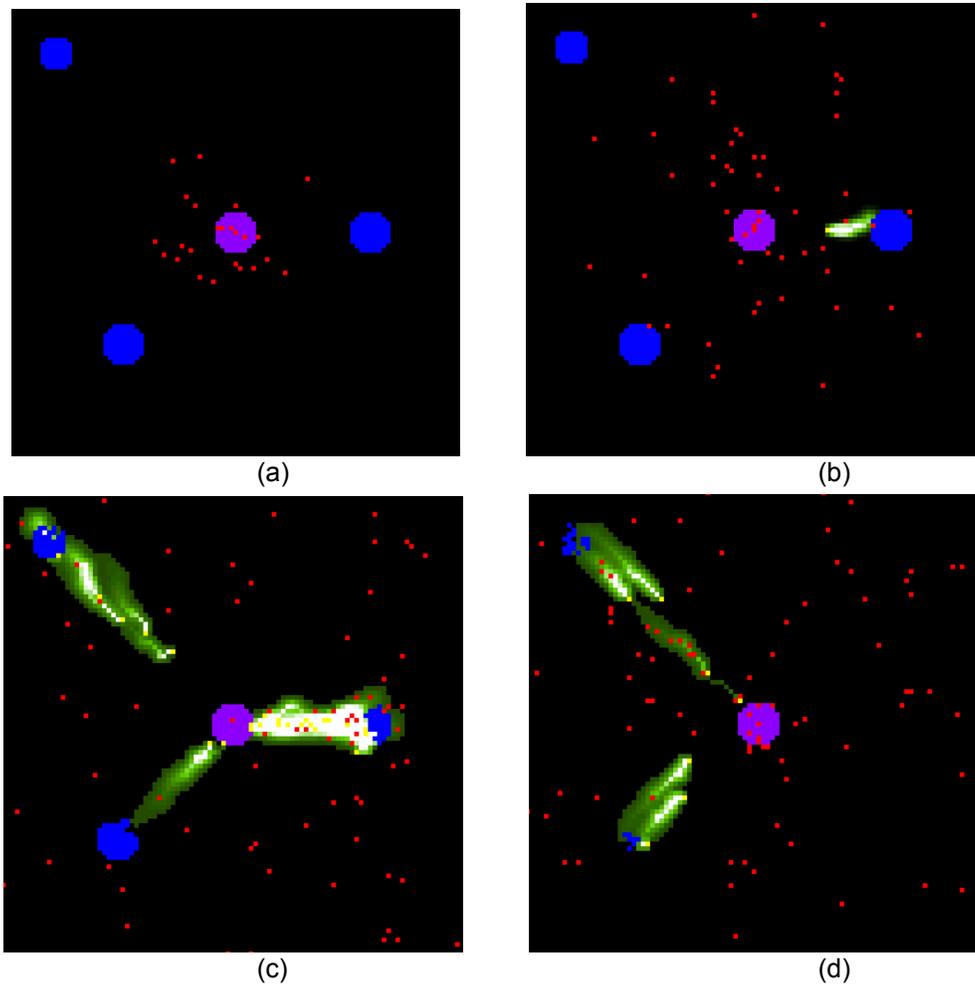
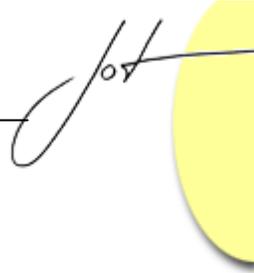


Figure 1 — Snapshots from an ant colony simulation.

Often, complex systems cannot be fully understood, precisely because we do not fully understand the component (or agent) interaction. For example, we might be able to identify the components involved in the New York Stock Exchange, yet we cannot accurately predict when we will have a bull market or a bear market, or when a market bubble will burst. In such situations, our knowledge of the interaction among the various components is not well understood.



2 COMPLEX SYSTEMS

The science of complex systems was first made popular by the Santa Fe Institute (SFI). This group of complex systems theorists and practitioners offer the following definition:

"Complexity refers to the condition of the universe which is integrated and yet too rich and varied for us to understand in simple common ... ways. We can understand many parts of the universe in these ways, but the larger more intricately related phenomena can only be understood by principles and patterns—not in detail."

This concept is important to us because the nature business and IT systems are becoming more complex. For almost a century our manufacturing model was reasonably stable. Rules for productivity, market dominance, and company success were well understood. However, the Information Age has turned previous "knowns" on their head. Today, no one can predict what or how rapidly new technologies will be developed. No one can accurately predict how supply chains will be affected or how customers will respond. A good business plan is only a guess.

Even small-to-medium-sized businesses are not insulated from this effect. Satellite communications, the Internet, and air transports commonly provide ways of rapidly moving corporate resources from place to place. All of us are now connected through a global market of online customers and suppliers, supply chain partnerships, and international franchise competition. As Susan Kelly and Mary Allison [1] suggest, businesses that don't understand the nature of complex system thinking and take advantage of it "will be at the mercy of an increasing number of sudden and unexpected shifts in the marketplace. As uncertainty grows exponentially with today's high rate of technological change and the fallout from it, so does the pressure of global markets." Many executives try to respond to this with yesterday's mindset and linear cause-and-effect thinking. Often, these responses will intensify an already downward spiral. Despite well-conceived plans and well-intended actions, a company that operates without complex-systems thinking will find itself unable to respond to the ever-increasing complexity of the business world.

Primary Issues

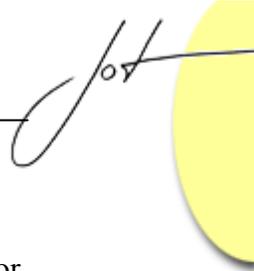
Common to any discussion of complex systems are several fundamental ideas.

- First and most basic among these is *agents*. In complex systems, these are the autonomous entities that interact to carry out their particular tasks.
- Another fundamental concept is that these agents are adaptive. That is, the agents must be able to react to their environment and possibly change their behavior based on what is learned.

- Complex systems are also characterized by *emergence*. Emergence is a coherent pattern that arises out of interactions among agents. For example, the process of an entire ant colony being fed was not programmed. It emerged from some very simple rule programmed into each ant. In other words, emergence is a by-product of individuals - not an choreographed result. Emergent results can be good as well as bad, and therefore must be considered when developing agent-based systems.
- Successful emergent systems often exist between order and chaos. As for any organism or organization, being orderly or chaotic all the time would result in death. However, the area in between is necessary for continued existence and fitness.
- Lastly, nature can teach us a lot about designing complex systems. It has been solving large combinatorial problems for billions of years. It makes sense, then, for us to consider notions such as parasitism, symbiosis, reproduction, genetics, mitosis, and survival-of-the fittest when developing our agent-based systems. For example, British Telecom is using the model of ants and pheromones in its call-routing network. Here, successful calls leave an equivalent of pheromones to guide future calls.

3 AGENTS AND EMERGENCE

Agents can work as noninteractive individuals or as a collective. When agents work as individuals with little or no interaction, what you get is just that: agents simply doing what they are asked to do. For example, a single "bot" agent sent out to find the cheapest airline fare, can be expected to simply return with the requested information. As a collective, however, something new and different can result - something that is *more* than the sum of the individual participants. The stock market, immune systems, and ant colonies are all examples of agents acting individually, yet from the interactions of these agents a new phenomenon arises. With the stock market, thousands of agents act independently to buy and sell shares of particular stocks and bonds. Yet from this independent behavior, an organism-like product called the stock market emerges. In other words, the rise and fall of the market is not controlled by a central process: it results from agents interacting. The stock market crash of '29 was a result of individual human agents - not a central controller. The crash of October '87 partly resulted from individual software agents that buy and sell securities according to programmed rules. The stock market, its crashes, temporary bubbles, dead-cat bounces, and all are more than the sum of the parts; it is an entity in its own right. Such entities are called *emergent* structures. Ant colonies are emergent structures that arise from individual ants acting interactively. The immune system emerges from the collective behavior of agents such as antigens, T cells, B cells, NK cells, immunoglobulins, lymph nodes, and the spleen. Other examples of emergent structures include families, organizations, societies, markets, flocks of birds,



and traffic jams. In IT systems, this can include supply-chain, scheduling, trading-floor, and e-commerce systems.

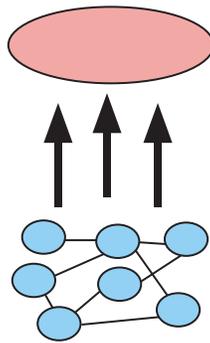


Figure 2 — Local interaction can give rise to global dynamics, creating a coherent structure.

Emergence

Emergence is the existence of a coherent pattern that arises out of interactions among agents.

The diagram in Fig. 2 illustrates this definition. Emergence embodies several properties.

- In emergent structures, agents organize into a whole that is greater than the sum of its parts. In other words, the parts alone do not result in emergent structures - their interaction is required. Single bot agents will not result in emergence; a multiagent environment such as shop floor operations systems handled by multiple interacting agents can have emergent properties.
- Rules that are almost absurdly simple can generate coherent, emergent phenomena. For example, each ant in the ant colony described earlier has some very simple rules, yet a well fed colony can emerge from these simple rules. A purchaser or supplier agent, too, can have simple rules resulting in a emergent inventory system.
- Instead of being designed from top down, most emergent systems emerge from the bottom up. A human engineer would tend to develop top down. This approach can be useful under some circumstances. However, in the world of living systems, most emerge from a population of simpler systems. Developing daily operational plans in a top-down manner work well for fairly predictable organizations. In less predictable environments, top-down plans can easily become obsolete in a manner of minutes; here a bottom-up agent-based approach could result in a very effective, emergent operation.

- Persistent emergent structures can serve as components of more complex emergent structures. In other words, hierarchies of emergent structures can be formed. This is how nature obtains scalability; any IT organization could employ the same mechanism.
- Agents and their emergent structure can form a two-way link. Agents can give rise to an emergent structure; the emergent structure can influence its component agents. For example, the stock market is an emergent structure of individuals buyers and sellers; yet, the rise and fall of such markets can in turn affect the buying and selling habits of individual participants.
- Emergent phenomena are typically persistent patterns with changing components. The birds in a flock or the cars in a traffic jam can change, yet the flock and traffic jam phenomena can remain. The buyers and suppliers in a company's supply chain change frequently, and the participants in a scheduling system can differ on a daily basis.
- Collections of agents can be homogeneous or heterogeneous. Emergence can occur due to the interaction of similar agents. More often, though, it occurs as a result of different kinds of agents that function in a society or ecosystem. Large organizations employ heterogeneity by specializing corporate resources using different roles and business units.

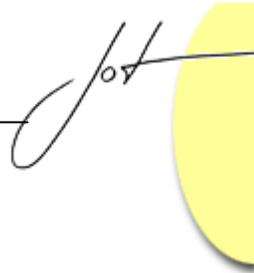
In the sections that follow, these properties will be explored in more detail.

Greater than the sum of its parts

Simple-minded reductionism states that the whole is simply the sum of its parts. Furthermore, each part can be studied in isolation. However, the parts alone can not produce emergence. Emergent structures also require the collective behavior and interaction of its components. Emergent structures, then, are a *process* - and the essence of the process is its form, not its parts. Families, organizations, societies, financial markets, schools of fish, and traffic jams are all examples of this phenomenon. A horde of noninteracting web spiders will not produce an emergent structure. However, when an ecosystem of supply-chain software agents can buy and sell goods and services over the web, a "supply web" can emerge [2]. The supply web, then, behaves like a financial market - which does give rise to an emergent structure. Agent-based modeling and bidding systems make this possible.

Simple agent rules can produce emergent structures

A common example of simple rules leading to emergence is that of a flock of birds. Here, each movement of a flock is so beautiful that it appears choreographed. Furthermore, the movements of the flock seem smoother than those of any one bird in the flock. Yet, the flock has no high-level controller or even a lead bird. Each bird follows a simple set of rules that it uses to react to birds nearby. In the flocking simulation of StarLogo (www.media.mit.edu/~starlogo), the birds obey only three rules:



1. If you are far away from other birds, head towards the nearest bird.
2. If you are about to crash into another bird, turn around.
3. Otherwise, fly in the same direction as the bird next to you.

Using these three simple rules, no one bird has a sense of an overall flock. The “bird in front” is merely a position of a given bird. It just happens to be there - and will be replaced by others in a matter of minutes. Flocks of birds are not the only things that work like this. Bee hives, ant colonies, freeway traffic, the Web, and the phenomenon of Silicon Valley are all examples of patterns that are determined by local component interaction, instead of a centralized authority. Complex behavior need not have complex roots.

Top-down versus bottom-up approaches

If you have ever heard classical music or watched a ballet, you will have no doubt that the performances have been orchestrated or choreographed. The centralized, or top-down, development of these kinds of performances is both obvious and necessary. Many of the products we use in everyday life require top-down engineering to be effective. However, most of the emergent phenomena we experience do not occur as a result of top-down efforts; instead, they are the result of decentralized, or bottom-up, processes. For example, the flock of birds mentioned above emerges without an organizer and behaves without a coordinator. So, too, does the applause that follows a classical concert or ballet.

Today, many resource providers and manufacturers are exploring the possibilities of employing a decentralized approach. Many of these organizations are already adopting solutions that will replace their central, globally optimized operations with a distributed, self-organizing, local one. John Holland, a professor at the University of Michigan, is fond of pointing out that New York City maintains a two-week supply of food with only locally made decisions. Companies, such as Boeing, John Deere & Company, and Detroit Edison, are beginning to do this.

Both centralized and decentralized approaches are useful techniques. Using one technique and not the other limits the possibilities of a system. Often, our human bias toward centralization precludes the consideration of decentralized solutions. For example, three-year old Rachel developed the theory that clouds rain when the thunder commands them. [3] At four, she developed a new theory: “The clouds get together at night and decide whether or not it should rain the next day.” People resist decentralization. When people see a pattern, they often assume a centralized control. This does not mean that centralized theories are wrong, it is just that they are not always appropriate:

- A central agent is a single point of failure that makes the system vulnerable to accident.
- Under normal operating conditions, a central agent can easily become a performance bottleneck.
- Even if it is adequately scaled for current operations, a central agent provides a boundary beyond which the system cannot be expanded.

- For software agents, it tends to attract functionality and code as the system develops, pulling the design away from the benefits of agents and in time becoming a large software artifact that is difficult to understand and maintain.

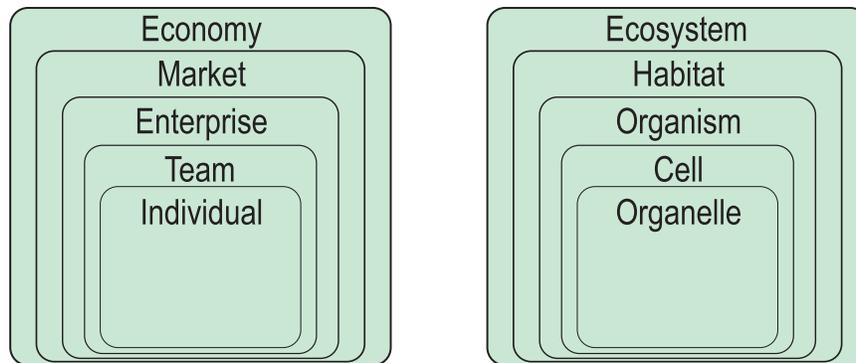


Figure 3 — Parallels between business and biology.

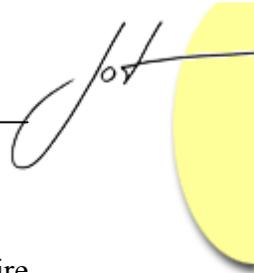
Emergent structures can themselves be components

One of the most difficult challenges for automated systems is scalability. Living systems provide some excellent examples of scaling 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, organelles can group to form cells, cells can aggregate to form organisms, and so on. In other words, living systems and their components emerge in a hierarchy of interlocking mechanisms. In the domain of human organization, similar hierarchies occur as illustrated in Fig. 3. Here, economies emerge from markets, which emerge from enterprises, and so on. [4]

Emergence provides the 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 from the one before and the one after it. Such a hierarchy of interlocking mechanisms is also an appropriate technique for automated agents.

Agents and their emergent structure can form a two-way link

Applause occurs when spectators join in what appears to be spontaneous synchronized clapping. There is no conductor that coordinates this. When everyone starts, the clapping is totally unorganized. Each person's tempo is wildly out of phase with the next person. Eventually groups of people begin clapping at the same tempo. People in the audience sense the emerging rhythms and adjust their clapping to join it. The emerging applause



rhythm grows even stronger and more people conform to it. Eventually, the entire audience is clapping in a synchronized pattern. This entire process can take place in a matter of seconds with even thousands of individuals.

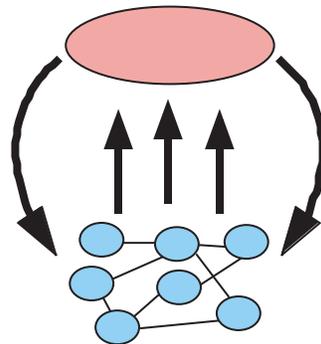


Figure 4 — Local interaction and global dynamics can influence each other.

We have been exploring how the local interaction and behavior of agents can produce global dynamics of emergent structures. In the example of spectator applause, the interaction of humans in the audience produced the dynamics of applause. However, there was another phenomenon occurring in this example: individually, the spectators adjusted their applause rhythm based on the applause that they heard. In other words, local interaction can give rise to global dynamics; and the global dynamics, in turn, can influence the local interaction.

Such an effect is evident in business systems, as well. The stock market both results from and affects the buyers and sellers of securities. The auto parts market setup as a joint venture of GM, Ford, and DaimlerChrysler will result from and affect the buyers and sellers of automotive parts. As illustrated in Fig. 4, emergent structures can be linked to their local agent interaction:

- This link influences the boundary conditions of the local agents.
- Local agents can then adjust to the presence of the global dynamics.
- Consequently, the conditions under which the agent behaves might change.

Emergent structures can have components that change

The birds in a flock or the cars in a traffic jam can change, yet the flock and traffic jam phenomena can remain. The same applies to the stock market and supply-chain webs. Just because an emergent structure exists and is stable does not mean that its components cannot change over time. Each of us replaces all the atoms in our body every three years, yet we are still considered to be the same being. Many Silicon Valley companies have more turnover than this, while still being recognized as the same organization. Change in

the underlying agent population is not required for emergence, but it is a common phenomenon.

Homogeneous and heterogeneous collections of agents

Some emergent structures consist of single kind of agent. For example, a flock of birds consists only of birds. Here, each agent plays the same role. Homogeneous agent collections can still have a single kind of agent, yet its agents can play different roles. For instance, an ant colony can have ants that play different roles. An ant can be a patrolling ant that guards the nest, a nest-maintenance worker, forager, brood-care worker, and so on. Furthermore, an ant can change its role depending on the requirements of the colony. For instance, a nest-maintenance worker can become a forager or a patrolling ant when the need for food or security becomes more important.

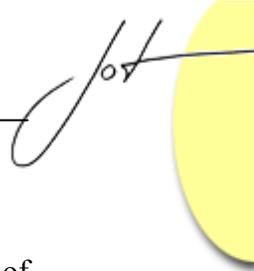
Heterogeneous collections of agents also play different roles because they contain different kinds of agents. The major difference is that in a heterogeneous collection, agents are different in both structure and behavior. For example, the immune system emerges from the collective behavior of various kinds of agents such as antigens, T cells, B cells, NK cells, immunoglobulins, lymph nodes, and the spleen. T cells and B cells not only play different roles, their structure and behavior is also different enough to be considered heterogeneous.

When we construct complex business systems, we need to think of agents as functioning as a society or ecosystem. In designing such systems, we need to consider how we can effectively employ homogeneous and heterogeneous agents.

4 CONCLUSION

When constructing agent systems, emergence is an important concept to consider. On the one hand, emergence is also something that can happen to you without your consent or predictive ability. This can be good or bad. On the other hand, as a system developer, you can try to “design in” the emergence that you want. In other words, you can try to design the agents in such a way that the desired structure emerges. In summary:

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



- Remember: a flock is not a big bird and a traffic jam is not just a collection of cars.

REFERENCES

- [1] Kelly, Susanne, and Mary Ann Allison: *The Complexity Advantage: How the Science of Complexity Can Help Your Business Achieve Peak Performance*, McGraw-Hill, New York, 1999.
- [2] Roy, Beau, "Using Agents to Make and Manage Markets Across a Supply Web," *Complexity*, 3:4, 1998, pp. 31-35.
- [3] Resnick, Mitchel, "Unblocking the Traffic Jams in Corporate Thinking," *Complexity*, 3:4, 1998, pp. 27-30.
- [4] Rothchild, Michael, *Bionomics: Economy as Ecosystem*, College Board, New York, 1995.

About the author



James J. Odell is a consultant, writer, and educator in the areas of object-oriented and agent-based systems, business reengineering, and complex adaptive systems. He has written four books on object orientation and has two books in progress on agent-based system design. His website is www.jamesodell.com.