Monitoring and Managing Intelligence in Distributed Systems

H. Van Dyke Parunak

NewVectors, division of TechTeam Government Solutions, Inc.
3520 Green Court Suite 250
Ann Arbor, MI 48105
van.parunak@newvectors.net

1 Abstract

Both the promise and the challenge of distributed systems often derive from emergent behavior, functionality that is not defined explicitly in the individual components of the system but results from their interactions. Effective use of these systems depends on techniques for monitoring and controlling such system-level behavior. A fruitful source of inspiration for such techniques is statistical mechanics, with its mature success in explaining the macro-level characteristics of matter from the interactions of atoms and molecules. This paper summarizes a number of such analogies that we have exploited in our work.

1 Introduction*

The phrase "intelligent agent systems" embodies an instructive ambiguity.

Some practitioners of the art view their task as constructing systems of individually intelligent agents, parsing the phrase,

```
((intelligent agent) systems).
```

Common examples of such systems are those based on negotiation or market metaphors.

Other researchers (our group included) focus on intelligence that arises at the system level from agents that no one would consider intelligent as individuals, parsing the phrase

```
(intelligent (agent systems)).
```

Swarm intelligence (Parunak 1997) is a classic example of such systems.

In (Intelligent (Agent Systems)), intelligence is not encoded in the agents, but emerges from their interactions with one another and with their environment. "Emerges" means that the system exhibits behavior that is not explicitly encoded in the components. Not every possible agent interaction results in intelligent system-level behavior, and as engineers of such systems, we need tools to monitor this emergent process.

A major source of inspiration for dealing with emergent behavior comes from statistical mechanics. This branch of physics seeks to explain how the macroscopic properties of matter arise from dynamics at the atomic and molecular level. It offers a rich array of concepts, paradigms, and intuitions that we have found useful in understanding the agent systems that we build.

This paper briefly summarizes four examples of emergence in multi-agent systems and how concepts from statistical mechanics can help monitor and manage the overall process. Table 1 summarizes the physical concepts and the problems to which we have applied them in multi-agent systems, and references some key publications where we develop these ideas in more detail.

Table 1: Physics Concepts for Distributed Systems

Concept from Statistical Mechanics	Application	References
Relaxation to Equilibrium	Convergence of anytime reasoner	(Parunak et al. 2005)
Entropy	Local detection of global state	(Brueckner and Parunak 2005; Parunak and Brueckner 2001)
Chaos	Sensitive dependence on initial conditions	(Parunak, Belding and Brueckner 2007)
Universality	Global behavior insensitive to local details	(Parunak, Brueckner and Savit 2004)

2 Equilibrium: Good Enough, Fast Enough

A physical system can be in or out of equilibrium, the position in its configuration space where the gradient of its potential energy is zero and thus where the sum of the forces and torques on its particles is zero. A stable equilibrium, where the second derivative of the potential energy is positive, is an attractor for the system, and the system's dynamics can be described in terms of its approach, or "relaxation," to one or more of these attractors.

The metaphor of a dynamical system out of equilibrium but relaxing toward an equilibrium configuration is a rich one for a highly distributed, decentralized system. MAS engineers recognize that they must design such a system so that its attractors correspond to the states they wish the system to approach (whether achievement goals or maintenance goals (Parunak 1999)). Instabilities in the environment may lead to shifting equilibria that the system never quite reaches, so that we must also be concerned with how rapidly the system approaches equilibrium and how close it can get in a given amount of time.

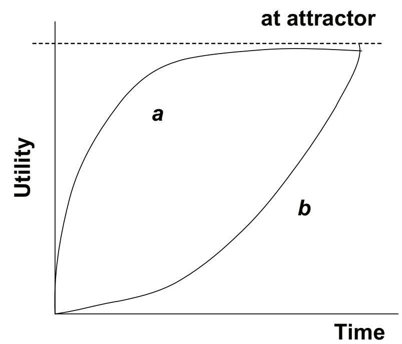

Figure 1: Notional relaxation functions. *a* = concave; *b* = convex.

Figure 1 shows two examples of how a system may approach an attractor. In general, an approach with a concave performance over time (monotone decreasing first derivative, curve *a*) is preferable to a convex one (monotone increasing first derivative, *b*) because it realizes most of its benefit early, before dynamic changes in the environment lead to a shift in the equilibrium.

Fortunately, many decentralized algorithms have a concave performance curve. Their aggregate dynamics are often captured by a finite-state Markov process, in which case the convergence is exponential (Goel 2004). We have shown good conformance to this exponential model in several distributed algorithms, including a distributed clustering algorithm (Figure 2). The top curve shows the convergence for a static collection of documents, while the others show convergence for configurations in which new documents are periodically added to the system at random locations in the emerging clustering. In the face of such disruption, the algorithm continues to converge (though to a lower asymptote). Notably, the exponent that indicates the convergence rate is the same across all cases.

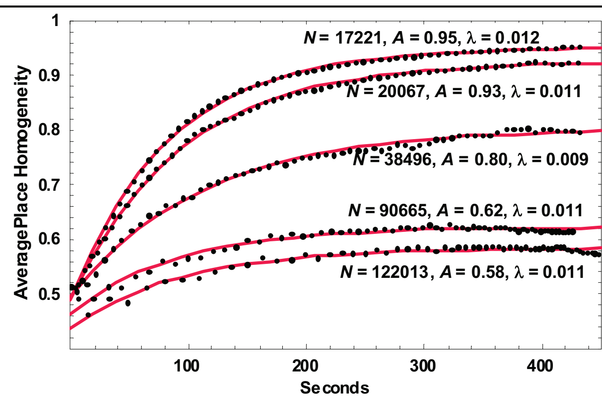

Figure 2: Distributed clustering with changing document population. All populations start at ~17221 and end at *N*. Fit is $y = A - e^{-\lambda t}$. λ for all curves is essentially the same.

The convergence of these systems can be derived explicitly from a relatively simple underlying model, an adaptive walk (Parunak et al. 2005). Consider a binary array $S \in \{0,1\}^N$ of length N. Initially, all elements of S are 0. At each time step,

1. Select an element of S at random.
2. If the element is currently 0, set it to 1 with probability p_{01}. If it is currently 1, set it to 0 with probability p_{10}. Note that p_{01} and p_{10} are independent. In particular, there is no requirement that they sum to 0.

The objective of this system is to maximize $N_1 = \Sigma S$, the number of elements of S that are set to 1.

Though simple, this model has essential features shared by more realistic systems.

- Each element of S represents an agent, and the array corresponds to the entire system of agents.
- The system objective is a global measure over the entire system.
- The agents do not have access to this global measure in making their decisions. In fact, in this very simple model, they do not consider the state of any other agent in making their decisions, but simply choose probabilistically. p_{01} reflects the probability that their local decision will end up advancing the global goal, while p_{10} reflects the likelihood that their local decision will in fact work against the objectives of the overall system.

This simple model predicts that $N_0 = |S| - N_1$, the number of 0's in S, will evolve according to $N_0 = \frac{1}{\lambda}\left(p_{10} + p_{01}e^{-\lambda t}\right)$, where $\lambda \equiv \left(p_{10} + p_{01}\right)/N$.

It offers several important benefits to the distributed systems engineer.

- It shows that a global system can converge, even under local decisions.
- It provides a way to estimate quantitatively the speed of convergence and the degree of performance that can be expected.
- It provides a baseline against which additional performance gains achieved by relaxing the requirement of local decisions can be assessed.

As an example of this last application, consider a set of UAV's responsible for surveillance of an area. We focus our attention on how rapidly the agents initially cover the territory that they must monitor, tracking the fraction of the area that has been seen as a function of time. A simple algorithm for this problem uses digital pheromones (Parunak, Brueckner and Sauter 2004; Parunak, Brueckner and Sauter 2002; Parunak, Purcell and O'Connell 2002). The environment is represented in the pheromone infrastructure as a lattice of square cells.

1. At regular intervals, each cell increments its level of attractive pheromone, propagates it to the eight neighboring cells, and evaporates it by a fixed proportion.
2. Every time a vehicle enters a new cell, it deposits repulsive pheromone in the cell corresponding to its current location., and zeros out the attractive pheromone in its current cell.
3. Each cell periodically evaporates (but does not propagate) the repulsive pheromone it contains by a fixed proportion.
4. Each vehicle moves to the neighboring cell for which difference (attractive pheromone – repulsive pheromone) has the greatest value.

Figure 3 shows three convergence curves for this system: an upper bound, the convergence predicted by the adaptive walk, and the actual experimental results.

The upper bound would be attained if at each time step, each vehicle could move immediately to a cell that has not yet been visited. Such a strategy is physically impossible, because it would permit vehicles to move directly between noncontiguous cells.

Our adaptive walk model is also unrealistic in this application, since it does not respect constraints on vehicle movement, but assumes that any vehicle can move immediately to any cell in the area, even one remote from its current location.

In spite of its unrealistic assumptions, the adaptive walk provides an excellent fit to the experimental data up to about 40% coverage. The inset shows how the adaptive walk rises slightly faster than the experiment, then falls below it.

The rapid rise reflects the idealistic assumptions of the adaptive walk: while the air vehicles are constrained by the need to move between contiguous cells, and so must often repeat coverage of cells that have already been seen, the adaptive walk can go directly to any cell.

Of more interest is the shortfall above 40% coverage. The adaptive walk slows as more and more of the area is covered, so that the random selection of the next site to visit frequently selects a site that has already been visited. The continued straight-line progress of our surveillance algorithm shows the effectiveness of the pheromone mechanism in improving over the random selection of the next site to visit. This improvement arises because pheromones

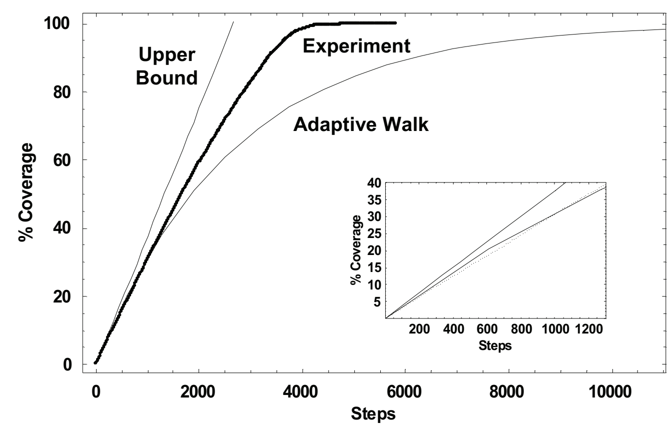

Figure 3: Comparison of adaptive walk with area surveillance.

reduce the locality of the decision process, in two ways. First, the propagation of attractive pheromone makes information from one cell available in a neighboring cell, reducing spatial locality and generating a gradient that guides the movement of vehicles. Second, the persistence of pheromone deposited by one vehicle for sensing by another reduces the temporal locality of decisions, enabling decisions at one point in time to take into account the results of previous decisions.

3 Entropy: Dealing with Disorder

One of the foundational principles of thermodynamics is the assertion that systems naturally tend to a state of maximum disorder, or entropy. Paradoxically, there are many physical examples of self-organization (Ball 1996), where system entropy appears to decrease over time. Self-organization is an important property for a distributed system, where we want to achieve global order without providing global information to the decision-making components. The concept of entropy is a promising tool for engineering self-organization.

The paradox posed by self-organizing physical systems is resolved by recognizing that these systems require an external source of usable energy. The flow of energy into the system represents a transfer of order from the outside environment into the system (or equivalently, a flow of disorder from inside the system to the environment). In other words, such systems reduce their internal entropy by increasing the entropy in their environment, effectively "leaking" entropy from structures where order is required into structures where disorder can be tolerated (Figure 4) (Parunak 1997).

We have demonstrated this entropy leakage in simple experiments in ant-inspired path formation (Parunak and Brueckner 2001), where the entropy of the ant decreases, balanced by an increase in the entropy of the pheromone field that the ant follows. Figure 5 shows the relation between the two entropies. The principle of an entropy leak is

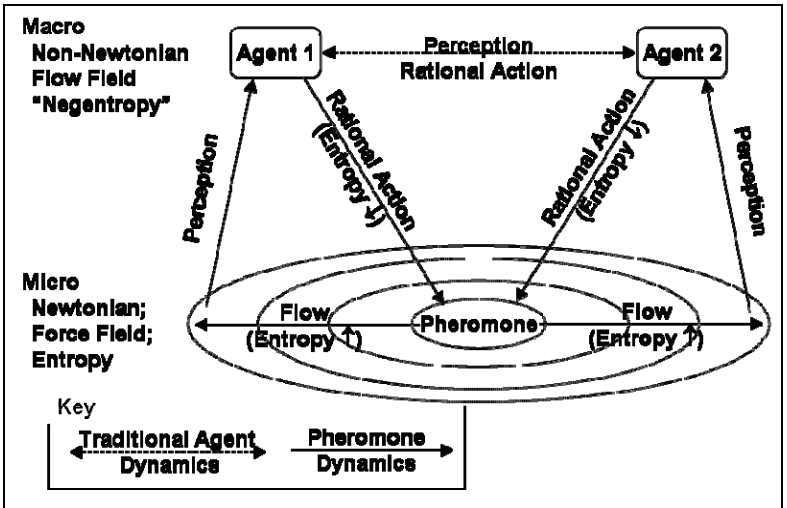

Figure 4: Entropy dynamics in multi-agent coordination

an important guideline for engineering self-organization in distributed information systems. The subsystem that is to become increasingly organized must be coupled to another into which it can leak entropy (such as a pheromone field). Otherwise the Second Law of thermodynamics will block the desired self-organization.

The previous example measured entropy over the global system. Entropy is also a useful way to think about a dilemma faced by a single agent in a distributed system. We often want individual agents to take action only when such an action is likely to improve the global state of the system. Otherwise, they simply contribute to thrashing that may actually slow down system convergence. A challenge to any agent seeking to make such a decision is that it has access only to local state information, not to the state of the global system.

The precise local information that is available to the agent varies from domain to domain. Two things are constant across domains. First, each agent must choose among alternative actions. Second, it modulates the probability with which it will choose each action on the basis of locally available information. Thus, in any domain, it makes sense to think about the distribution of probabilities associated with the set of options open to the agent. This distribution is compactly summarized by the Option Set Entropy (OSE). Briefly, if the probability of the ith option is p_i, then the OSE is $-\Sigma p_i \ln p_i$. (Sometimes it is useful to normalize this value by the log of the number of cases, to obtain a value in [0,1] that is independent of the size of the problem.) The OSE assumes its maximum when all the probabilities are the same, which means that locally available information offers no guidance to the agent. In this case, the agent can only thrash, and may be better advised to skip a turn. When the OSE is low, the environment is giving a clear signal to the agent, which can act with some confidence that it is not wasting its time.

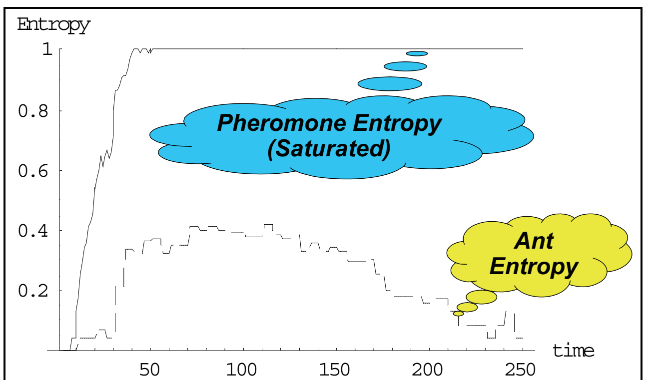

Figure 5: Entropy leak. The ant reduces its entropy in approaching the goal (lower curve) by leaking entropy to the pheromone field (upper curve).

We illustrate this approach in a distributed graph coloring problem, a useful model for resource allocation problems (where nodes correspond to tasks, links between nodes indicate tasks that execute concurrently, and colors correspond to resources). The degree of conflict (DoC) of a colored graph is the proportion of edges that join nodes of the same color. If individual nodes choose their colors uniformly randomly from G alternatives, this value will be $1/G$. So the global DoC multipled by G gives a useful indicator of the global effectiveness of the behaviors of individual agents, taking values less than 1 when the algorithm under study performs better than random.

In this example, we study the behavior of a simple distributed algorithm (whose details are not relevant to our discussion) developed at the Kestrel Institute to study the effect of delayed information (Fitzpatrick and Meertens 2001). The effectiveness of this algorithm varies with the configuration of the colored graph. Figure 6 shows the global degree of conflict achieved by this algorithm over a range of graph sizes N and number of colors G. The figure shows that the algorithm works best for a "sweet spot" in

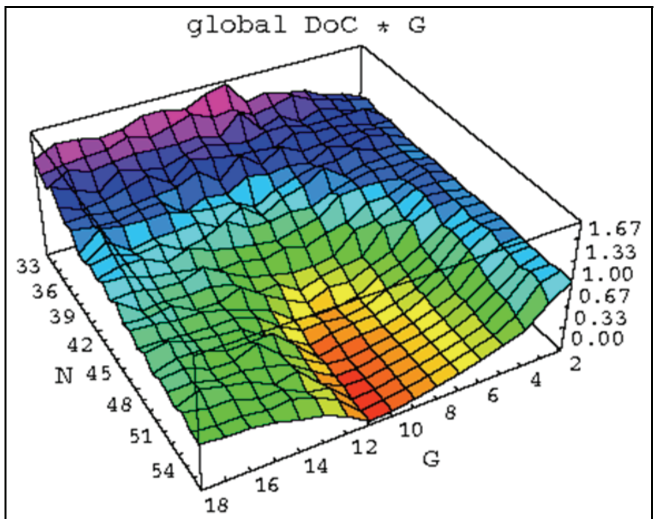

Figure 6: Normalized global Degree of Conflict for a distributed graph coloring algorithm.

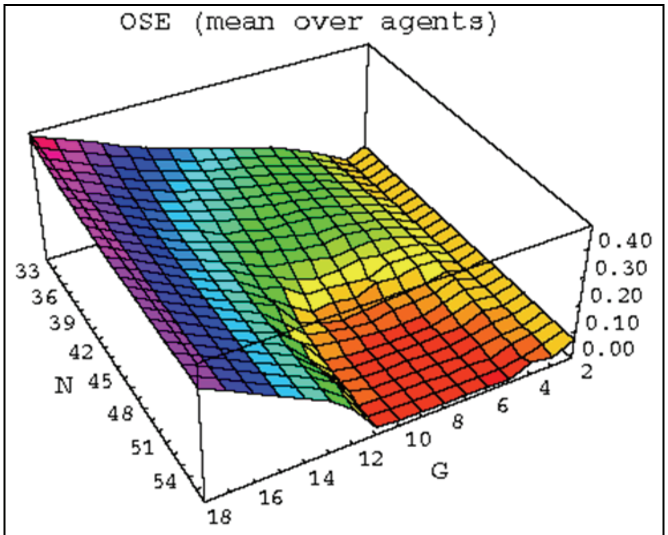

Figure 7: Mean Option Set Entropy for distributed graph coloring

graph configuration space (graphs larger than 45 nodes, with between 6 and 12 colors), and has difficulty with graphs that lie outside these parameters.

One can imagine an enhanced version of this algorithm in which agents use the prescribed algorithm in the sweet spot, and vary their behavior outside of it. Hard-coding the boundaries of the sweet spot is not satisfactory. In a realistic resource allocation problem, the location of the sweet spot may not be known in advance, and it may depend on factors other than number of nodes and available resources. We would like individual agents (nodes) to be able to detect locally when the system is in the sweet spot. However, the global degree of conflict is a global parameter, not readily accessible to individual agents.

Figure 7 shows the mean OSE over the same configuration space. When the system is within the sweet spot, agents see a much lower OSE, and can usefully follow the simple algorithm. When the OSE increases, the environment gives them less guidance, and nodes may need to use more sophisticated measures.

Figure 7 is only suggestive of the potential of the OSE, since it plots the mean value across all agents, itself a global value. However, agents can compute their local OSE and compare it with their neighbors to obtain a local view of the system's convergence. We are conducting further research to determine the potential of this metric for guiding individual agent behavior.

4 Chaos: Too Clever by Half

The potential of computers to simulate physical processes led to a profound appreciation in the last half of the twentieth century for the potential for chaotic behavior in nonlinear systems. This behavior is characterized by a sensitive dependence on initial conditions, so that starting a system with initial conditions very close together in state space may lead to an arbitrarily large divergence in their end conditions. For example, consider the logistic equation, $p_{t+1} = p_t + 3p_t(1 - p_t)$. The 1000th iterate of this equation from $p_0 = 0.01$ is[1] 0.136739. However, if we add 10^{-13} to p_0, the result is 0.0423537, a difference of more than three times.

The interactions among agents in a distributed system are almost always nonlinear. Nonlinearity may be as simple as a threshold or a rule-based decision algorithm. As a result, even when a system is run twice with similar starting conditions, it may yield very different answers. Or it may not, if the environment is sufficiently constraining. When we design and manage distributed systems, we must be aware of the potential for chaotic divergence of trajectories. Here are two examples of the implications of this potential.

First, while simulation is a valuable tool in engineering distributed systems, we should be suspicious of the results of any single run. We need to sample many trajectories to see whether they diverge, or whether the environment will constrain the system to repeatable behavior. In many cases, If many trajectories beginning at slightly different locations cluster closely together, we can rely on the simulation to be predict the behavior of the system. Otherwise, caution is in order.

Sampling sufficient trajectories can be challenging. Imagine $n + 1$ entities simulated in discrete time. At each step, each entity interacts with one of the other n. Thus at time t its interaction history $h(t)$ is a string in n^t. Its behavior is a function of $h(t)$. This toy model generalizes many domains, including predator-prey systems, combat, innovation, diffusion of ideas, and disease propagation.

During a run of length τ, each entity will experience one of n^τ possible histories. (This estimate is of course worst case, since domain constraints may make many of these histories inaccessible.) The population of $n + 1$ entities will sample $n + 1$ of these possible histories. It is often the case that the length of a run is orders of magnitude larger than the number of modeled entities ($\tau >> n$).

A single run is hardly a representative sample of the space of possible histories. Multiple runs with different random seeds is only a partial solution, since each run adds only one set of possible interactions out of the n^τ possible histories. For large populations and scenarios that permit multiple interactions on the part of each agent, the number of runs needed to sample the possible alternative interactions thoroughly can quickly become prohibitive. In one of our applications, $n \sim 50$ and $\tau \sim 10{,}000$, so the proportion of the space of possible entity histories actually sampled by a single run is vanishingly small. We would need on the order of τ runs to generate a meaningful sample, and executing that many runs is out of the question.

[1] Readers may obtain different results due to differences in the numerical processors, algorithms, or system libraries.

Polyagents (Parunak and Brueckner 2006) offer a novel approach to sampling multiple possible interactions of different entities in the course of a single run. Each entity is represented by a single persistent avatar and a population of transient ghost agents. The ghosts run into the future and generate a digital pheromone field that summarizes the avatar's range of possible behaviors. The avatar manages the population of ghosts and interprets their findings. Up to a constant, the pheromone field generated by an entity's ghosts reflects the probability that the entity will be at a location in space-time. As the ghosts of other entities interact with this field, they are exploring the product of the possible trajectories for their entity with the possible trajectories of the first entity. If each entity generates k ghosts, the total number of interactions sampled in each time step is $O(k^n)$, far more than the single interaction in a conventional simulation, and much more reliable for estimating the overall behavior of the system. When an entity's ghosts bunch together, we can expect that the entity's trajectory will be relatively constrained by the environment. When they diverge, we are cautioned that its behavior will be less predictable.

The polyagent approach is one example of many distributed algorithms that iterate to look into the future. Such prediction introduces a second example for the potential of trajectory divergence to impact a distributed system. A short prediction may be reliable, but trajectory divergence may lead to longer-range predictions that are no better than random guesses. Thus the dynamics of the system may impose a "prediction horizon," beyond which we cannot reliably see.

We have demonstrated this effect in a simple predator-prey game (Parunak, Belding and Brueckner 2007). Two randomly distributed populations of avatars, c Cowards and r Rambos[2], are situated in a toroidal arena k units on a side (a k x k square whose top and bottom are connected, as are its left and right sides). Cowards flee from Rambos, while Rambos approach Cowards. Each avatar deposits a digital pheromone in the environment, tracking its location. At each time step, each avatar samples the pheromone of avatars of the other side in its environment, and probabilistically decides whether it has engaged the adversary. The higher the pheromone concentration, the more likely an engagement is. If an engagement takes place, the Coward avatar dies with probability p (1.0 in our current experiments). Rambo avatars are immortal. When a Coward avatar dies, a new one is born at a random location, keeping the population constant.

Each avatar generates ghosts (one per time step) to guide its movement. The ghosts execute a random walk starting with their avatar, for a number of steps (the lookahead of the prediction). The ghosts determine the likelihood of an encounter by sampling the other side's pheromone. When a Coward ghost and a Rambo ghost encounter each other, with probability p (again 1.0 in current experiments) they kill each other, and their avatars are notified of the location where the encounter took place. (Rambo ghosts, unlike Rambo avatars, are not immortal.)

Rambo ghosts live for 100 time steps (but only report their deaths within 10 steps of the avatar). The lifetime of Coward ghosts is the main independent variable in our experiment, and represents the distance into the future that the Coward looks ahead.

Periodically, each avatar takes a step. The interval between steps is the maximum of the Rambo and Coward ghost lifetimes. Each Rambo avatar wants to find Coward avatars, so it is attracted toward locations where its ghosts have died in encounters with Coward ghosts. It takes one step in the direction of its "guidance vector," a weighted sum of unit vectors from its current location to each of its ghosts that have died since it last moved.

Coward avatars want to avoid Rambo avatars. Each Coward avatar computes a weighted sum of unit vectors to its dead ghosts and takes a step in the opposite direction.

We expect performance of Cowards first to increase as the lookahead increases, and then to decrease as it passes the prediction horizon. Success for a Coward means evading the Rambos, and thus experiencing fewer casualties. Since runs are of a fixed length, we use the total number of Coward avatar casualties as a performance figure. Thus our dependent variable can be interpreted in terms of the death rate of Cowards.

For a run of 500 steps with ten Cowards and five Rambos on a 10x10 arena, Figure 8 plots the number of Coward deaths at prediction horizons ranging from 0 to 100. As expected, the number of deaths at first drops rapidly with increasing lookahead, from nearly 110 with 0 lookahead to about 35 with a lookahead of 4. Then it climbs rapidly and asymptotes around 90. Cowards can usefully predict and avoid threats for horizons on the order of 3-5, but then the future becomes increasingly murky.

Prediction typically alternates with action. A system pre-

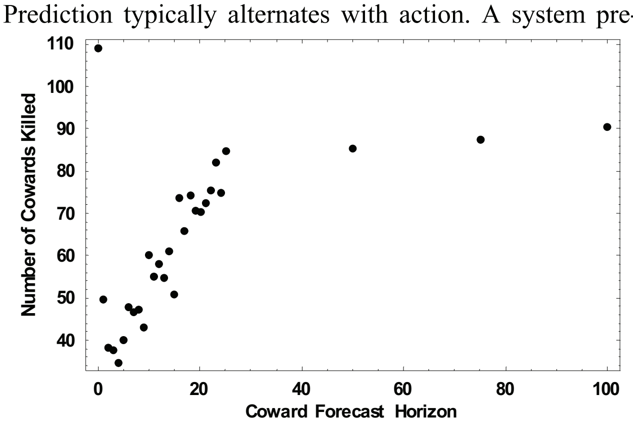

Figure 8. Coward deaths as function of lookahead, 10x10 arena.

[2] We use this term as the semantic opposite to "Coward," to denote an agent that aggressively seeks to engage its adversaries.

dicts in order to choose its next action. If we do not look far enough ahead, our actions may have undesirable consequences. If we try to look past the prediction horizon, our actions will also have undesirable consequences (because the predictions will be unreliable), and in addition we will waste computation time generating the longer-range prediction. Awareness of the prediction horizon will let us know how far we can reliably look ahead, and when it is time to take action.

5 Universality: When Intelligence Doesn't Matter

Most research in multi-agent systems (MAS's) emphasizes mechanisms for the internal reasoning of individual agents, usually justified by theoretical elegance or simple tests of a few agents. In systems of interacting agents, aspects of overall system behavior are often relatively independent of the algorithms used by individual agents.

Universality in physics is a useful metaphor for understanding how structural features can neutralize details of individual behavior, and for guiding further research into the interplay of agent and environment. These insights are important in making engineering decisions concerning MAS's. For instance, more complex agents in general are associated with higher knowledge engineering costs and may require more complex hardware, higher power budgets, and more complex input data to execute than simpler agents. Understanding the environmental circumstances under which a simpler agent can achieve the same functionality will enable more efficient designs.

In physics, universality describes the behavior of the critical exponents associated with a continuous phase transition. A phase transition is a mathematical singularity experienced by a system as some parameter varies. It is continuous if the discontinuity is apparent in the first or higher-order derivative of the observed parameter with respect to external variables, but not in the value of the parameter itself.

An example is the net magnetization of iron at the Curie temperature t_c (760° Celsius). Below t_c, iron is ferromagnetic: when exposed to a magnetic field, it becomes magnetized in the direction of the field, and this magnetization remains after the field is removed. Above t_c, it becomes paramagnetic: its magnetization is proportional to the applied field, and vanishes when the field is zero. In such transitions, near t_c, all relevant physical quantities (e.g., magnetic susceptibility, specific heat, isothermal compressibility) exhibit power-law behavior, varying as $|t-t_c|^{\beta}$ (Figure 9). The exponent β depends on the physical quantity, and may be positive (the quantity vanishes at t_c) or negative (it diverges). Remarkably, the value of the critical exponents is relatively independent of the material being studied. For example, the exponent for the liquid-gas coexistence curve is the same (about 0.33) for Ne, A, Kr, Xe, N_2, O_2, CO, and CH_4, although these substances differ widely in atomic weight, molecular structure, and the details of their electrochemical interactions.

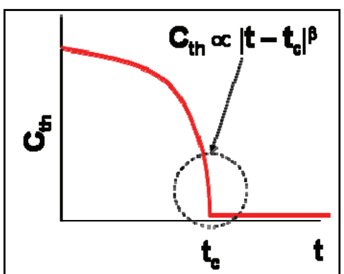

Figure 9: Power-law behavior of continuous phase transition

Two physical systems will exhibit universality if their interactions have the same spatial dimensionality and interaction symmetry. These structural similarities in their interactions neutralize the detailed characteristics of the molecules and of their interactions, so that the difference between (say) Ne and CH_4 no longer affects the behavior of the system near the critical point.

We extend universality from its simple, precise meaning in statistical mechanics to include any system of interacting elements whose qualitative or quantitative system-level behavior includes characteristics that are invariant under changes in the individual behavior and detailed interaction of the elements. In these cases we say that differences in individual behavior are *neutralized* by the interactions. A universality hierarchy can be defined, ranging from highly qualitative similarities, through those of an intermediate character, down to the simplest and most quantitative (exemplified by critical exponents in phase transitions).

Universality can be viewed as a manifestation of Ashby's Law of Requisite Variety (Ashby 1958): the greater the variety of actions in a control system, the greater the variety of disturbances it is able to compensate. Conversely (Principia Cybernetica 2003), the amount of appropriate selection that a controller can perform is limited by the information available. A physical system's spatial dimensions and the symmetry of the Hamiltonian limit the information to which molecules can respond, restricting the scope of their actions. Similarly, if agent interactions limit the information available to agents or their ability to respond non-randomly, any sophistication above the level appropriate to the environmental information will not make a difference in the system's behavior, and in fact will introduce noise that can limit system efficiency.

Several key characteristics of the phenomenon in physical systems make it non-trivial, and pose analogous challenges in applying the concept to MAS's.

- Universality affects some, but not all, characteristics of a system. An MAS may exhibit universality in some system-level characteristics, but not others.
- Universality arises in some, but not all, regimes of a system's operation. An MAS that exhibits universality in one regime may not do so in another. There are analogs to phase transitions in computational systems (Hogg, Huberman and Williams 1996; Monasson et al. 1999), including MAS's (Savit et al.

2002). The physical analogy suggests that the vicinity of these transitions is a natural place to explore for universality, but our examples show instances of universality that are not near phase shifts.

- Universality defines a set of equivalence classes, not a single class. There may be classes of agent architectures within which universality obtains, but between which it does not.

MAS's are subject to complex and sometimes chaotic behavior (Parunak 1999; Parunak, Brueckner and Sauter 2002). Such extreme sensitivity to small differences in initial conditions seems at first glance to contradict our emphasis that system-level behavior can be universal across different agent behaviors. One distinction is that sensitivity to initial conditions describes the evolution of a *single* system through time, while universality compares *multiple* systems either synchronically or diachronically. For example, the geometry of a system's attractor may exhibit universality, while a specific trajectory on the attractor may be exhibit extreme sensitivity.

MAS's are intrinsically complex, and it is not currently possible to address their universality analytically. Our study begins with disciplined reports of observed universality, clarifying the structure and operating regime of the system, identifying the aspects of behavior that are observed to be universal, and discussing the possible interaction effects that lead to universality and how differences among individual agent behaviors may be neutralized by or immune to these effects.

We have observed universality in both qualitative and quantitative forms in a number of experiments with distributed resource allocation. A qualitative example compares the minority game (Savit, Manuca and Riolo 1999) and distributed graph coloring (Fitzpatrick and Meertens 2001). Though very different in their structure and algorithms, both systems exhibit the same three behavioral regimes.

1. In the low-information regime, agents exhibit herding behavior that leads to thrashing, resulting in resource overloading worse than under random allocation.
2. In the high-information regime, agents are unable to process the information presented to them, and effectively make random choices.
3. In between, in both systems, there is an optimal level of information enabling the community as a whole to achieve better than random patterns of allocation.

The common feature to both systems is that they are boundedly rational. Any such reasoner will be best suited for a particular amount of information (Takashina and Watanabe 1996). If the information available is greater than this level, the mechanism will be overwhelmed and give answers essentially no better than random. If it is lower than this level, the mechanism will not be able to break the symmetry among the agents, resulting in herding and thrashing. We hypothesize that these regimes will *not* appear for truly optimizing decision-makers, since an optimal decision by definition uses all of the relevant information available. However, such processes are rarely encountered in practical real-world domains. Real-time constraints impose time limits that effectively turn even an optimizer into a bounded rationalizer when confronted by large enough amounts of information.

6 Conclusion

Some MAS researchers have argued that emergent behavior is harmful, and agent behaviors should be constrained to ensure that it does not occur (Moore and Wright 2003; Wooldridge and Jennings 1998). Our approach is just the opposite. Experience suggests that emergence wants to happen. This tendency has two implications.

First, a system constrained tightly enough to exclude emergence is likely to be extremely brittle. Emergence results from nonlinear interactions, and an emergence-proof system must have impoverished interactions and much greater centralization than would otherwise be necessary.

Second, the pervasive nature of emergence suggests that it is robust and adaptive, characteristics that we desire for our engineered systems.

Thus, our approach is not to seek to exclude emergence, but to develop tools to model, monitor, and manage it. A major source of inspiration for our efforts has been statistical physics, with its long history of reasoning about the relation between micro and macro perspectives of systems. Physical concepts such as relaxation to equilibrium, entropy, chaotic behavior, and universality have proven powerful tools for analyzing and operating highly distributed, decentralized computational systems. The examples summarized in this paper only scratch the surface in suggesting the fruitfulness of this perspective.

7 Acknowledgments

This paper summarizes work done by the Agent-Based and Complex systems (ABC) group at NewVectors and its predecessors, with notable contributions from Ted Belding, Steve Brophy, Sven Brueckner, Rainer Hilscher, Bob Matthews, John Sauter, and Bob Savit.

8 References

Ashby, W. R. 1958. "Requisite variety and its implications for the control of complex systems." *Cybernetica* 1(2): 83-99.

Ball, P. 1996. *The Self-Made Tapestry: Pattern Formation in Nature*. Princeton, NJ: Princeton University Press.

Brueckner, S. and Parunak, H. V. D. 2005. Information-Driven Phase Changes in Multi-Agent Coordination. In *Proceedings of Workshop on Engineering Self-Organizing Systems (ESOA, at AAMAS 2005)*, Utrecht, Netherlands, Springer. 104-119. http://www.newvectors.net/staff/parunakv/AAMAS03InfoPhaseChange.pdf.

Fitzpatrick, S. and Meertens, L. 2001. Soft, Real-Time, Distributed Graph Coloring using Decentralized, Synchronous, Stochastic, Iterative-Repair, Anytime Algorithms: A Framework. Palo Alto, CA: Kestrel Institute. ftp://ftp.kestrel.edu/pub/papers/fitzpatrick/eMergeANT-001.pdf.

Goel, S. 2004. Three Approaches to Finite Markov Chains. Ithica, NY: Cornell University. http://www.cam.cornell.edu/~sharad/papers/FMC.pdf.

Hogg, T., Huberman, B. A. and Williams, C. 1996. "Phase Transitions and the Search Problem." *Artificial Intelligence* 81: 1-15. http://www.parc.xerox.com/istl/groups/iea/abstracts/PhaseTransitions/specialIssueIntro.html.

Monasson, R., Zecchina, R., Kirkpatrick, S., Selman, B. and Troyansky, L. 1999. "Determining computational complexity from characteristic 'phase transitions'." *Nature* 400(July 8): 133-137. http://xxx.lanl.gov/ps/cond-mat/9910080.

Moore, D. and Wright, W. 2003. Emergent Behaviours Considered Harmful. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003)*, Melbourne, Australia. 1070-1071.

Parunak, H. V. D. 1997. "'Go to the Ant': Engineering Principles from Natural Agent Systems." *Annals of Operations Research* 75: 69-101. http://www.newvectors.net/staff/parunakv/gotoant.pdf.

Parunak, H. V. D. 1999. From Chaos to Commerce: Practical Issues and Research Opportunities in the Nonlinear Dynamics of Decentralized Manufacturing Systems. In *Proceedings of Second International Workshop on Intelligent Manufacturing Systems*, Leuven, Belgium, Katholieke Universiteit Leuven. k15-k25. http://www.newvectors.net/staff/parunakv/ims99.pdf.

Parunak, H. V. D., Belding, T. C. and Brueckner, S. 2007. Prediction Horizons in Polyagent Models. In *Proceedings of Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS07)*, Honolulu, HI. 930-932. www.newvectors.net/staff/parunakv/AAMAS07PH.pdf.

Parunak, H. V. D. and Brueckner, S. 2001. Entropy and Self-Organization in Multi-Agent Systems. In *Proceedings of The Fifth International Conference on Autonomous Agents (Agents 2001)*, Montreal, Canada, ACM. 124-130. www.newvectors.net/staff/parunakv/agents01ent.pdf.

Parunak, H. V. D. and Brueckner, S. 2006. Modeling Uncertain Domains with Polyagents. In *Proceedings of International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, Hakodate, Japan, ACM. http://www.newvectors.net/staff/parunakv/AAMAS06Polyagents.pdf.

Parunak, H. V. D., Brueckner, S. and Sauter, J. 2002. ERIM's Approach to Fine-Grained Agents. In *Proceedings of First International Workshop on Radical Agent Concepts (WRAC 2002)*, Greenbelt, MD. http://www.newvectors.net/staff/parunakv/WRAC2001.pdf.

Parunak, H. V. D., Brueckner, S. and Sauter, J. 2004. Digital Pheromones for Coordination of Unmanned Vehicles. In *Proceedings of Workshop on Environments for Multi-Agent Systems (E4MAS 2004)*, New York, NY, Springer. 246-263. http://www.newvectors.net/staff/parunakv/E4MAS04_UAVCoordination.pdf.

Parunak, H. V. D., Brueckner, S. and Savit, R. 2004. Universality in Multi-Agent Systems. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*, New York, NY, ACM. 930-937. http://www.newvectors.net/staff/parunakv/AAMAS04Universality.pdf.

Parunak, H. V. D., Brueckner, S. A. and Sauter, J. 2002. Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles. In *Proceedings of First International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, Bologna, Italy, ACM. 449-450. www.newvectors.net/staff/parunakv/AAMAS02ADAPTIV.pdf.

Parunak, H. V. D., Brueckner, S. A., Sauter, J. A. and Matthews, R. 2005. Global Convergence of Local Agent Behaviors. In *Proceedings of Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS05)*, Utrecht, The Netherlands, ACM. 305-312. http://www.newvectors.net/staff/parunakv/AAMAS05Converge.pdf.

Parunak, H. V. D., Purcell, M. and O'Connell, R. 2002. Digital Pheromones for Autonomous Coordination of Swarming UAV's. In *Proceedings of First AIAA Unmanned Aerospace Vehicles, Systems,Technologies, and Operations Conference*, Norfolk, VA, AIAA. www.newvectors.net/staff/parunakv/AIAA02.pdf.

Principia Cybernetica. 2003. "Law of Requisite Variety." Retrieved 10 December, 2003, from http://pespmc1.vub.ac.be/ASC/LAW_VARIE.html.

Savit, R., Brueckner, S. A., Parunak, H. V. D. and Sauter, J. 2002. "Phase Structure of Resource Allocation Games." *Physics Letters A* 311: 359-364. http://arxiv.org/pdf/nlin.AO/0302053.

Savit, R., Manuca, R. and Riolo, R. 1999. "Adaptive Competition, Market Efficiency, and Phase Transitions." *Physical Review Letters* 82(10): 2203-2206.

Takashina, T. and Watanabe, S. 1996. The Locality of Information Gathering in Multiagent Systems. In *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS-96)*. 461. http://homepage.mac.com/t_takashina/paper-dir/ICMAS96_takashina.pdf.

Wooldridge, M. J. and Jennings, N. R. 1998. Pitfalls of Agent-Oriented Development. In *Proceedings of 2nd Int. Conf. on Autonomous Agents (Agents-98)*, Minneapolis, MN. 385-391. http://citeseer.nj.nec.com/wooldridge98pitfalls.html.