Concurrent Modeling of Alternative Worlds with Polyagents

H. Van Dyke Parunak[1], Sven Brueckner[1]

[1]Altarum Institute, 3520 Green Court, Suite 300, Ann Arbor, MI 48105 USA
+1 734 302 4600
{van.parunak, sven.brueckner}@altarum.org

Abstract. Agent-based modeling is a powerful tool for systems modeling. Instantiating each domain entity with an agent captures many aspects of system dynamics and interactions that other modeling techniques do not. However, an entity's agent can execute only one trajectory per run, and does not sample the alternative trajectories accessible to the entity in the evolution of a realistic system. Averaging over multiple runs does not capture the range of individual interactions involved. We address these problems with a new modeling entity, the *polyagent*, which represents each entity with a single persistent *avatar* supported by a swarm of transient *ghosts*. Each ghost interacts with the ghosts of other avatars through digital pheromone fields, capturing a wide range of alternative trajectories in a single run that can proceed faster than real time.

1 Introduction

The fundamental entity in an agent-based model (ABM), the agent, corresponds to a discrete entity in the domain. The fundamental operator is interaction among agents. The fundamental entity in an equation-based model (EBM) [20] is a system observable, and the fundamental operator is its evolution (e.g., by a differential equation).

ABM's often map more naturally to a problem than do EBM's, are easier to construct and explore, and provide more realistic results [9, 19], but have a shortcoming. Observables in an EBM are often averages across agents, and implicitly capture the range of agent variation (at an aggregate level). By contrast, the agent representing an entity in an ABM can execute only one trajectory per run of the system, and does not capture the alternative trajectories that the entity might have experienced. Averaging over multiple runs still does not capture the range of individual interactions involved.

A new modeling construct, the *polyagent*, represents each entity with a single persistent *avatar* and multiple transient *ghosts*. Each ghost interacts with the ghosts of other avatars through digital pheromones, exploring many alternative trajectories in a single run that can proceed faster than real time for many reasonable domains. We have used this approach in several applications. This paper articulates the polyagent as an explicit modeling construct and provides some guidance concerning its use.

Section 2 reviews the sampling challenge sampling in ABM. Section 3 proposes the polyagent as an answer to this challenge, and Section 4 compares it with other technology. Section 5 reports on polyagent systems we have constructed. Section 6 discusses what these examples teach us and considers directions for research on polyagents. Section 7 concludes.

2 The Challenge of Modeling Multi-Agent Interactions

Imagine $n + 1$ entities 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.

It would be convenient if a few runs of such a system told us all we need to know, but this is not likely to be the case, for three reasons.

1. We may have imperfect knowledge of the agents' internal states or details of the environment (for example, in a predator-prey system, the carrying capacity of the environment). If we change our assumptions about these unknown details, we can expect the agents' behaviors to change.
2. The agents may behave non-deterministically, either because of noise in their perceptions, or because they use a stochastic decision algorithm.
3. Even if the agents' reasoning and interactions are deterministic and we have accurate knowledge of all state variables, nonlinearities in decision mechanisms or interactions can result in overall dynamics that are formally chaotic, so that tiny differences in individual state variables can lead to arbitrarily large divergences in agent behavior. A nonlinearity can be as simple as a predator's hunger threshold for eating a prey or a prey's energy threshold for mating.

An EBM typically deals with aggregate observables across the population. In the predator-prey example, such observables might be predator population, prey population, average predator energy level, or average prey energy level, all as functions of time. No attempt is made to model the trajectory of an individual entity.

An ABM must explicitly describe the trajectory of each agent. In a given run of a predator-prey model, depending on the random number generator, predator 23 and prey 14 may or may not meet at time 354. If they do meet and predator 23 eats prey 14, predator 52 cannot later encounter prey 14, but if they do not meet, predator 52 and prey 14 might meet later. If predator 23 happens to meet prey 21 immediately after eating prey 14, it will not be hungry, and so will not eat prey 21, but if it did not first encounter prey 14, it will consume prey 21. And so forth. A single run of the model can capture only one set of many possible interactions among the agents.

In our general model, 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 ($\tau >> n$).

Multiple runs with different random seeds is only a partial solution. Each run only samples one set of possible interactions. 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 the application described in Section 4.3, $n \sim 50$ and $\tau \sim 10{,}000$, so the sample 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.

We need a way to capture the outcome of multiple possible interactions among agents in a few runs of a system. Polyagents are one solution to this problem.

3 The Polyagent Modeling Construct

A polyagent represents a single domain entity. It consists of a single *avatar* that manages the correspondence between the domain and the polyagent, and a swarm of *ghosts* that explore alternative behaviors of the domain entity.

The *avatar* corresponds to the agent representing an entity in a conventional multi-agent model of the domain. It persists as long as its entity is active, and maintains state information reflecting its entity's state. Its computational mechanisms may range from simple stigmergic coordination to sophisticated BDI reasoning.

Each avatar generates a stream of *ghost agents*, or simply *ghosts*. Ghosts typically have limited lifetime, dying off after a fixed period of time or after some defined event to make room for more ghosts. The avatar controls the rate of generation of its ghosts, and typically has several ghosts concurrently active.

Ghosts explore alternative possible behaviors for their avatar. They interact with one another stigmergically, through a digital pheromone field, a vector of scalar values ("pheromone flavors") that is a function of both location and time. That is, each ghost chooses its actions stochastically based on a weighted function of the strengths of the various pheromone flavors in its immediate vicinity, and deposits its own pheromone to record its presence. A ghost's "program" consists of the vector of weights defining its sensitivity to various pheromone flavors.

Having multiple ghosts multiplies the number of interactions that a single run of the system can explore. Instead of one trajectory for each avatar, we now have one trajectory for each ghost. If each avatar has k concurrent ghosts, we explore k trajectories concurrently. But the multiplication is in fact greater than this.

The digital pheromone field supports three functions [1, 11]:

1. It *aggregates* deposits from individual agents, fusing information across multiple agents and through time. In some of our implementations of polyagents, avatars deposit pheromone; in other, ghosts do. Aggregation of pheromones enables a single ghost to interact with multiple other ghosts at the same time. It does not interact with them directly, but only with the pheromone field that they generate, which is a summary of their individual behaviors.
2. It *evaporates* pheromones over time. This dynamic is an innovative alternative to traditional truth maintenance in artificial intelligence. Traditionally, knowledge bases remember everything they are told unless they have a reason to forget something, and expend large amounts of computation in the NP-complete problem of reviewing their holdings to detect inconsistencies that result from changes in the domain being modeled. Ants immediately begin to forget everything they learn, unless it is continually reinforced. Thus inconsistencies automatically remove themselves within a known period.

3. It *propagates* pheromones to nearby places, disseminating information.

This third dynamic (propagation) enables each ghost to sense multiple other agents. If n avatars deposit pheromones, each ghost's actions are influenced by up to n other agents (depending on the propagation radius), so that we are exploring in effect $n*k$ interactions for each entity, or n^2*k interactions overall. If individual ghosts deposit pheromones, the number of interactions being explored is even greater, on the order of k^n. Of course, the interactions are not played out in the detail they would be in a conventional multi-agent model. But our empirical experience is that they are reflected with a fidelity that is entirely adequate for the problems we have addressed.

Pheromone-based interaction not only multiplies the number of interactions that we are exploring, but also enables extremely efficient execution. In one application, we support 24,000 ghosts concurrently, faster than real time, on a 1 GHz Wintel laptop.

The avatar can do several things with its ghosts, depending on the application.

- It can activate its ghosts when it wants to explore alternative possible futures, modulating the rate at which it issues new ghosts to determine the number of alternatives it explores. It initializes the ghosts' weight vectors to define the breadth of alternatives it wishes to explore.
- It can evolve its ghosts to learn the best parameters for a given situation. It monitors the performance of past ghosts against some fitness parameter, and then breeds the most successful to determine the parameters of the next generation.
- It can review the behavior of its swarm of ghosts to produce a unified estimate of how its own behavior is likely to evolve and what the range of likely variability is.

4 Comparison with Other Paradigms

Our polyagent bears comparison with several previous multi-agent paradigms and two previous uses of the term (Table 1).

Polyagents are distinct from the common use of agents to model different functions of a single domain entity. For example, in ARCHON [21], the domain entity is an electrical power distribution system, and individual agents represent different functions or perspectives required to manage the system. In a polyagent, each ghost has the same function: to explore one possible behavior of the domain entity. The plurality of ghosts provides, not functional decomposition, but a range of estimates of alternative behaviors.

Table 1. Comparing the Polyagent with Other Technologies

	Multiple agents per domain entity	Avatar/Ghost dualism	Parallel search of alternative behaviors	Parallel search of multiple possible interactions
Polyagent	X	X	X	X
Functional agents	X			
Evolutionary computation	X		X	
Fictitious play	X		X	
Ant colony optimization	X		X	X
Kijima's polyagents				
Polyagent therapies				

Many forms of evolutionary computation [4] allow multiple representatives of a single entity to execute concurrently, to compare their fitness. In these systems, each agent samples only one possible series of interactions with other entities. Pheromone-based coordination in the polyagent construct permits each ghost to adjust its behavior based on many possible alternative behaviors of other entities in the domain.

Similarly, the multiple behaviors contemplated in fictitious play [7] take place against a static model of the rest of the world.

Like the polyagent, ant-colony optimization [2] uses pheromones to integrate the experiences of parallel searchers. The polyagent's advance is the notion of the avatar as a single point of contact for the searchers representing a single domain entity.

The term "polyagent" is a neologism for several software agents that collectively represent a domain entity and its alternative behaviors. The term is used in two other contexts that should not lead to any confusion. In medicine, "polyagent therapy" uses multiple treatment agents (notably, multiple drugs combined in chemotherapy). Closer to our domain, but still distinct, is the use of the term by K. Kijima [5] to describe a game-theoretic approach to analyzing the social and organizational interactions of multiple decision-makers. For Kijima, the term "poly-agent" makes sense only as a description of a system, and does not describe a single agent. In our approach, it makes sense to talk about a single modeling construct as "a polyagent."

5 Examples of Polyagents

We discovered polyagents by reflecting on several applications that we have constructed and observing their common features.

5.1 Factory Scheduling

Our first application of polyagents was to real-time job-shop scheduling [1]. We prototyped a self-organizing multi-agent system with three species of agents: processing resources, work-pieces, and policy agents. Avatars of processing resources with different capabilities and capacities and avatars of work-pieces with dynamically changing processing needs (due to re-work) jointly optimize the flow of material through a complex, high-volume manufacturing transport system. In this application, only the avatars of the work-pieces actually deploy ghosts. The policy agents and avatars of the processing resources (machines) are single agents in the traditional sense.

In a job shop, work-pieces interact with one another by blocking access to the resources that they occupy, and thus delaying one another. Depending on the schedule, different work-pieces may interact, in different orders. Polyagents explore the space of alternative routings and interactions concurrently in a single model.

Work-piece avatars currently loaded into the manufacturing system continuously deploy ghosts that emulate their decision processes in moving through various decision points in the manufacturing process. Each of these decisions is stochastic, based on the relative concentration of attractive pheromones in the neighborhood of the next decision point. These pheromones are actually deposited by the policy agents that try to optimize the balance of the material flow across the transport network, but they are modulated by the ghosts. Thus, an avatar's ghosts modulate the pheromone field to which the avatar responds, establishing an adaptive feedback loop into the future.

The avatars continuously emit ghosts that emulate their current decision process. The ghosts travel into the future without the delay imposed by physical transport and processing of the work-pieces. These ghosts may find the next likely processing step and wait there until it is executed physically, or they may emulate the probabilistic outcome of the step and assume a new processing state for the work-piece they are representing. In either case, while they are active, the ghosts contribute to a pheromone field that reports the currently predicted relative load along the material flow system. When ghosts for alternative work-pieces explore the same resource, they interact with one another through the pheromones that they deposit and sense.

By making stochastic decisions, each ghost explores an alternative possible routing for its avatar. The pheromone field to which it responds has been modulated by all of the ghosts of other work-pieces, and represents multiple alternative routings of those work-pieces. Thus the ghosts for each work-piece explore both alternative futures for that work-piece, and multiple alternative interactions with other work-pieces.

Policy agents that have been informed either by humans or by other agents of the desired relative load of work-pieces of specific states at a particular location in turn deposit attractive or repulsive pheromones. Thus, through a local adaptive process, multiple policy agents supported by the flow of ghost agents adapt the appropriate levels of pheromone deposits to shape the future flow of material as desired.

By the time the avatar makes its next routing choice, which is delayed by the physical constraints of the material flow through the system, the ghosts and the policy agents have adjusted the appropriate pheromones so that the avatar makes the "right" decision. In effect, the policy agents and the ghosts control the avatar as long as they can converge on a low-entropy pheromone concentration that the avatar can sample.

5.2 Path Planning for Robotic Vehicles

Two pressures require that path planning for robotic vehicles be an ongoing activity. 1) The agent typically has only partial knowledge of its environment, and must adapt its behavior as it learns by observation. 2) The environment is dynamic: even if an agent has complete knowledge at one moment, a plan based on that knowledge becomes less useful as the conditions on which it was based change. These problems are particularly challenging in military applications, where both targets and threats are constantly appearing and disappearing.

In the DARPA JFACC program, we approached this problem by imitating the dynamics that ants use in forming paths between their nests and food sources [8]. The ants search stochastically, but share their discoveries by depositing and sensing nest and food pheromone. Ants that are searching for food deposit nest pheromone while climbing the food pheromone gradient left by successful foragers. Ants carrying food deposit food pheromone while climbing the nest pheromone gradient. The initial pheromone trails form a random field, but quickly collapse into an optimal path as the ants interact with one another's trails.

The challenge in applying this algorithm to a robotic vehicle is that the algorithm depends on interactions among many ants, while a vehicle is a single entity that only traverses its path once. We use a polyagent to represent the vehicle (in our case, an aircraft) whose route needs to be computed [12, 17]. As the avatar moves through the battlespace, it continuously emits a swarm of ghosts, whose interactions mimic the ant dynamics and continuously (re)form the path in front of the avatar. These ghosts seek targets and then return to the avatar. They respond to several digital pheromones:

- *RTarget* is emitted by a target.
- *GNest* is emitted by a ghost that has left the avatar and is seeking a target.
- *GTarget* is emitted by a ghost that has found a target and is returning to the avatar.
- *RThreat* is emitted by a threat (e.g., a missile battery).

Ideally, the digital pheromones are maintained in a distributed network of unattended ground sensors dispersed throughout the vehicle's environment, but they can also reside on a central processor, or even on multiple vehicles. In addition, we provide each ghost with *Dist*, an estimate of how far away the target is.

In general, ghosts are attracted to RTarget pheromone and repelled from RThreat pheromone. In addition, before they find a target, they are attracted to GTarget pheromone. Once they find a target, they are attracted to GNest pheromone. A ghost's movements are guided by the relative strengths of these quantities in its current cell and each neighboring cell in a hexagonal lattice. It computes a weighted combination of these factors for each adjacent cell and selects stochastically among the cells, with probability proportionate to the computed value.

Each ghost explores one possible route for the vehicle. The avatar performs two functions in overseeing its swarm of ghosts.

1. It *integrates* the information from the several ghosts in their explorations of alternative routes. It observes the GTarget pheromone strength in its immediate vicinity, and guides the robot up the GTarget gradient. GTarget pheromone is deposited only by ghosts that have found the target, and its strength in a given cell reflects the number of ghosts that traversed that cell on their way home from the target. So the aggregate pheromone strength estimates the likelihood that a given cell is on a reasonable path to the target.
2. It *modulates* its ghosts' behaviors by adjusting the weights that the ghosts use to combine the pheromones they sense. Initially, all ghosts used the same hand-tuned weights, and differences in their paths were due only to the stochastic choices they made in selecting successive steps. When the avatar randomly varied the weights around the hand-tuned values, system performance improved by more than 50%, because the ghosts explored a wider range of routes. We then allowed the avatar to evolve the weight vector as the system operates, yielding an improvement nearly an order of magnitude over hand-tuned ghosts [16].

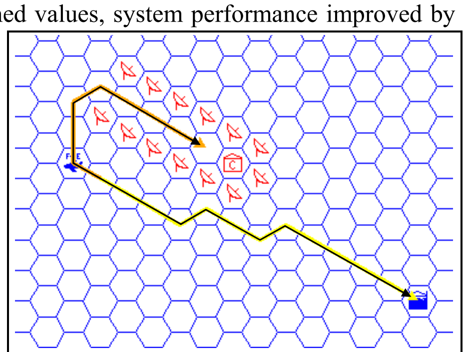

Fig. 1. Gauntlet Routing Problem

We tested this system's ability to route an aircraft in simulated combat [12]. In one example, it found a path to a target through a gauntlet of threats (Fig. 1). A centralized route planner seeking an optimal path by integrating a loss function and climbing the resulting gradient was unable to solve this problem without manually introducing a waypoint at the gauntlet's entrance. The polyagent succeeded because some of the ghosts, moving stochastically, wandered into the gauntlet, found their way to the target, and then returned, laying pheromones that other ghosts could reinforce.

Another experiment flew multiple missions through a changing landscape of threats and targets. The figure of merit was the total surviving strength of the Red and Blue forces. In two scenarios, the aircraft's avatar flew a static route planned on the basis of complete knowledge of the location of threats and targets, without ghosts. The routes differed based on how closely the route was allowed to approach threats. A third case used ghosts, but some threats were invisible until they took action during the simulation. Fig. 2 compares these three cases. The polyagent's ability to deal with partial but up-to-date knowledge both inflicted more damage on the adversary and offered higher survivability than preplanned scripts based on complete information.

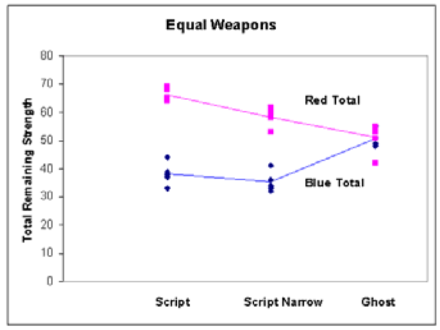

Fig. 2. Real-Time vs. Advance Planning.—"Script" is a conservative advance route based on complete knowledge. "Script narrow" is a more aggressive advance route. "Ghost" is the result when the route is planned in real time based on partial knowledge

Route planning shows how a polyagent's ghosts can explore alternative behaviors concurrently, and integrate that experience to form a single course of action. Since only one polyagent is active at a time, this work does not draw on the ability of polyagents to manage the space of possible interactions among multiple entities.

5.3 Characterizing and Predicting Agent Behavior

The DARPA RAID program [6] focuses on the problem of characterizing an adversary in real-time and predicting its future behavior. Our contribution to this effort [15] uses polyagents to evolve a model of each real-world entity (a group of soldiers known as a fire team) and extrapolate its behavior into the future. Thus we call the system "the BEE" (Behavior Evolution and Extrapolation).

Fig. 3 is an overview of the BEE process. Ghosts live on a timeline indexed by τ that begins in the past at the insertion horizon and runs into the future to the prediction horizon. τ is offset with respect to the current time t. The timeline is divided into discrete "pages," each representing a successive value of τ. The avatar inserts the ghosts at the insertion horizon. In our current system, the insertion horizon is at $\tau - t = -30$, meaning that ghosts are inserted into a page representing the state of the world 30 minutes ago. At the insertion horizon, the avatar samples each ghost's rational and emotional parameters (desires and dispositions) from distributions to explore alternative personalities of the entity it represents. The avatar is also responsible for estimating its entity's goals (using a belief network) and instantiating them in the environment as pheromone sources that constrain and guide the ghosts' behavior. In estimating its entity's goals and deliberately modulating the distribution of ghosts, the avatar reasons at a higher cognitive level than do the pheromone-driven ghosts.

Each page between the insertion horizon and $\tau = t$ ("now") records the historical state of the world at its point in the past, represented as a pheromone field generated by the avatars (which at each page know the actual state of the entity they are modeling). As ghosts move from page to page, they interact with this past state, based on their behavioral parameters. These interactions mean that their fitness depends not just on their own actions, but also on the behaviors of the rest of the population, which is also evolving. Because τ advances faster than real time, eventually $\tau = t$ (actual time). At this point, the avatar evaluates each of its ghosts based on its location compared with the actual location of its corresponding real-world entity.

The fittest ghosts have three functions.

1. The avatar reports personality of the fittest ghost for each entity to the rest of the system as the likely personality of the corresponding entity. This information enables us to characterize individual warriors as unusually cowardly or brave.
2. The avatar breeds the fittest ghosts genetically and reintroduces their offspring at the insertion horizon to continue the fitting process.
3. The fittest ghosts for each entity run past the avatar's present into the future. Each ghost that runs into the future explores a different possible future of the battle, analogous to how some people plan ahead by mentally simulating different ways that a situation might unfold. The avatar analyzes the behaviors of these different possible futures to produce predictions of enemy behavior and recommendations for friendly behavior. In the future, the pheromone field with which the ghosts interact is generated not by the avatars, but by the ghosts themselves. Thus it integrates the various possible futures that the system is considering, and each ghost

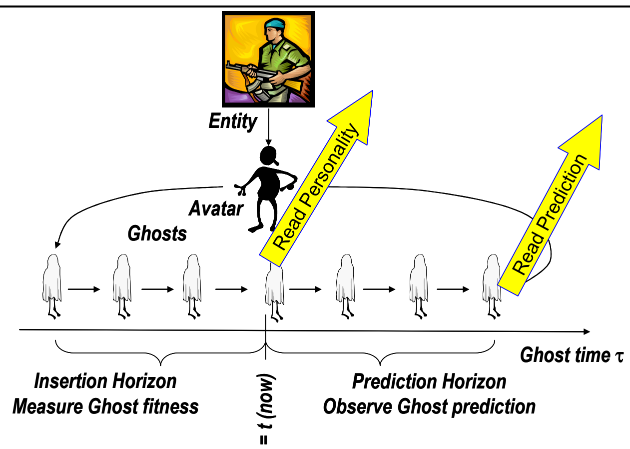

Fig. 3. Behavioral Emulation and Extrapolation. Each avatar generates a stream of ghosts that sample the personality space of its entity. They evolve against the entity's recent observed behavior, and the fittest ghosts run into the future to generate predictions

is interacting with this composite view of what other entities may be doing.

The first and third functions are analogous to the integrating function of the avatars in route planning, while the second is analogous to the modulation function.

This model has proven successful both in characterizing the internal state of entities that we can only observe externally, and in predicting their future behavior. [15] details the results of experiments based on multiple wargames with human participants. We can detect emotional state of entities as well as a human observer, but faster. Our prediction of the future of the battle is also comparable with that of a human, and much better than a "guessing" baseline based on a random walk.

6 Discussion

These projects reflect several common features that deserve recognition as a new and useful modeling construct, and that we now articulate as a "polyagent."

- Multiple agents (the ghosts) concurrently explore alternative possible behaviors of the domain entity being modeled.
- The ghosts interact through a digital pheromone field that permits simultaneous reasoning about the multiple possible interactions among the domain entities.
- A single, possibly more complex agent (the avatar), *modulates* the swarm of ghosts, controlling the number of ghosts, the rate at which they are introduced, and the settings and diversity of their behavior. In our most sophisticated cases (route planning and agent fitting), the avatar evolves the ghosts.
- The avatar also *integrates* the behaviors of its several ghosts (either directly or by observing the pheromones they deposit) to produce a single higher-level report on the domain entity's likely behavior.

Our use of polyagents involves a fair amount of art, and is motivated by their successful application in multiple applications. Theoretical work is needed to make the technique more rigorous. One challenging question is the legitimacy of merging pheromones of multiple ghosts representing alternative futures for agent A of one type into a single field that then guides the behavior of agent B of a different type. This process is qualitatively distinct from the merger of pheromone deposits from multiple agents living in the *same world* to form an optimized path guiding other agents of the *same type* (the heart of conventional ant optimization). The multiple-worlds version enables B to explore concurrently its possible interactions with multiple alternative realizations of A, but we need to justify this process more formally.

The strength of a pheromone field depends, *inter alia*, on the frequency with which agents visit various locations. Thus it may be viewed as a probability field describing the likelihood of finding an agent of a given type at a given location. If those agents are ghosts representing alternative futures of an entity's trajectory, the probability field may be interpreted in terms of the likelihood of different future states. Table 2 suggests several parallels between this perspective on polyagents and quantum physics [3]. In the spirit of our earlier work applying metaphors from theoretical physics to understanding multi-agent systems [10, 13, 14, 18], we will use concepts from quantum mechanics to provide an intellectual and formal model for engineering polyagent systems and interpreting their behavior.

Like quantum wave models, polyagents explore multiple possible behaviors and interactions. Unlike wave functions, they can do so predictively. We can configure polyagents to model what will happen in the future based on current policies, then use the avatars' summary of what will happen to guide changes to those policies.

Table 2. Quantum Physics and Polyagents

Quantum Physics	Polyagents
Duality between (single, localized) particle and (distributed) wave function	Duality between (single, localized) avatar and (distributed) swarm of ghosts
Interactions among wave functions' amplitude fields model interactions among particles	Ghosts' pheromone fields can be interpreted as probability densities that model interactions of agents
Wave function captures a range of possible behaviors	Swarm of ghosts captures a range of possible behaviors
Observation collapses the wave function to a single behavior	Avatar interprets the aggregate behavior of the ghosts and yields a single prediction of behavior

7 CONCLUSION

One strength of ABM's over EBM's is that they capture the idiosyncracies of each entity's trajectory. In complex domains, this strength is also a weakness, because any single set of trajectories is only a sample from a large space of possible trajectories. Possible interactions among the agents explode combinatorially, making this space much too large to explore thoroughly by repeated experiments.

Polyagents can sample multiple interactions in a single run. An avatar mediates between the real-world entity being modeled and a swarm of ghosts that sample its alternative possible trajectories. The avatar may employ sophisticated cognitive reasoning, but the ghosts are tropistic, interacting through digital pheromone fields that they deposit and sense in their shared environment. The avatar modulates the generation of ghosts, and interprets their aggregate behavior to estimate its entity's likely behavior.

We have applied this system to scheduling and controlling manufacturing jobs, planning paths for unpiloted air vehicles through a complex adversarial environment, and characterizing the internal state of fighting units from observations of their outward behavior, and then projecting their likely behavior into the future to form predictions. Empirically, the polyagent functions well, but invites theoretical work on the interpretation of multiple ghosts interacting with a pheromone field that represents multiple alternative realizations of other entities. Several parallels with quantum physics suggest the latter discipline may be a guide in developing a more formal model.

Acknowledgements

This material is based on work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Nos. F3062-99-C-0202 and NBCHC040153. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA or the

Department of Interior-National Business Center (DOI-NBC). Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

References

[1] S. Brueckner. *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*. Dr.rer.nat. Thesis at Humboldt University Berlin, Department of Computer Science, 2000. http://dochost.rz.hu-berlin.de/dissertationen/brueckner-sven-2000-06-21/PDF/Brueckner.pdf.

[2] M. Dorigo and T. Stuetzle. *Ant Colony Optimization*. Cambridge, MA, MIT Press, 2004.

[3] R. Feynman and A. R. Hibbs. *Quantum Mechanics and Path Integrals*. McGraw-Hill, 1965.

[4] C. Jacob. *Illustrating Evolutionary Computation With Mathematica*. San Francisco, Morgan Kaufmann, 2001.

[5] K. Kijima. Why Stratification of Networks Emerges in Innovative Society: Intelligent Poly-Agent Systems Approach. *Computational and Mathematical Organization Theory*, 7(1 (June)):45-62, 2001.

[6] A. Kott. Real-Time Adversarial Intelligence & Decision Making (RAID). 2004. http://dtsn.darpa.mil/ixo/programdetail.asp?progid=57.

[7] T. J. Lambert, III, M. A. Epelman, and R. L. Smith. A Fictitious Play Approach to Large-Scale Optimization. *Operations Research*, 53(3 (May-June)), 2005.

[8] H. V. D. Parunak. 'Go to the Ant': Engineering Principles from Natural Agent Systems. *Annals of Operations Research*, 75:69-101, 1997. http://www.altarum.net/~vparunak/gotoant.pdf.

[9] H. V. D. Parunak, R. Savit, and R. L. Riolo. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In *Proceedings of Multi-agent systems and Agent-based Simulation (MABS'98)*, Paris, FR, pages 10-25, Springer, 1998. http://www.altarum.net/~vparunak/mabs98.pdf.

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

[11] H. V. D. Parunak. Making Swarming Happen. In *Proceedings of Swarming and Network-Enabled C4ISR*, Tysons Corner, VA, ASD C3I, 2003. http://www.altarum.net/~vparunak/MSH03.pdf.

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

[13] H. V. D. Parunak, S. Brueckner, and R. Savit. Universality in Multi-Agent Systems. In *Proceedings of Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004)*, New York, NY, pages 930-937, IEEE, 2004. http://www.altarum.net/~vparunak/AAMAS04Universality.pdf.

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

[15] H. V. D. Parunak and S. A. Brueckner. Extrapolation of the Opponent's Past Behaviors. In A. Kott and W. McEneany, Editors, *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, Chapman and Hall/CRC Press, Boca Raton, FL, 2006.

[16] J. A. Sauter, R. Matthews, H. V. D. Parunak, and S. Brueckner. Evolving Adaptive Pheromone Path Planning Mechanisms. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS02)*, Bologna, Italy, pages 434-440, 2002. www.altarum.net/~vparunak/AAMAS02Evolution.pdf.

[17] J. A. Sauter, R. Matthews, H. V. D. Parunak, and S. A. Brueckner. Performance of Digital Pheromones for Swarming Vehicle Control. In *Proceedings of Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Utrecht, Netherlands, pages 903-910, 2005. http://www.altarum.org/~vparunak/AAMAS05SwarmingDemo.pdf.

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

[19] N. M. Shnerb, Y. Louzoun, E. Bettelheim, and S. Solomon. The importance of being discrete: Life always wins on the surface. *Proc. Natl. Acad. Sci. USA*, 97(19 (September 12)):10322-10324, 2000.

[20] J. Sterman. *Business Dynamics*. New York, NY, McGraw-Hill, 2000.

[21] T. Wittig. *ARCHON: An Architecture for Multi-agent Systems*. New York, Ellis Horwood, 1992.