Real-Time Agent Characterization and Prediction

H. Van Dyke Parunak, Sven Brueckner, Robert Matthews, John Sauter, Steve Brophy
NewVectors LLC
3520 Green Court, Suite 250
Ann Arbor, MI 48105 USA
+1 734 302 4684
{van.parunak, sven.brueckner, robert.matthews, john.sauter, steve.brophy}@newvectors.net

ABSTRACT

Reasoning about agents that we observe in the world is challenging. Our available information is often limited to observations of the agent's external behavior in the past and present. To understand these actions, we need to deduce the agent's internal state, which includes not only rational elements (such as intentions and plans), but also emotive ones (such as fear). In addition, we often want to predict the agent's future actions, which are constrained not only by these inward characteristics, but also by the dynamics of the agent's interaction with its environment. BEE (Behavior Evolution and Extrapolation) uses a faster-than-real-time agent-based model of the environment to characterize agents' internal state by evolution against observed behavior, and then predict their future behavior, taking into account the dynamics of their interaction with the environment.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – *parameter learning.* I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence– *multiagent systems.*

General Terms

Algorithms, Measurement, Experimentation.

Keywords

Plan recognition, plan inference, evolution, prediction, emotion, BDI, swarm intelligence, digital pheromones, dynamics.

1. INTRODUCTION

Reasoning about agents that we observe in the world must integrate two disparate levels. Our observations are often limited to the agent's external behavior, which can frequently be summarized numerically as a trajectory in space-time (perhaps punctuated by actions from a fairly limited vocabulary). However, this behavior is driven by the agent's internal state, which (in the case of a human) may involve high-level psychological and cognitive concepts such as intentions and emotions. A central challenge in many application domains is reasoning from external observations of agent behavior to an estimate of their internal state. Such reasoning is motivated by a desire to predict the agent's behavior.

This problem has traditionally been addressed under the rubric of "plan recognition" or "plan inference." Work to date focuses almost entirely on recognizing the rational state (as opposed to the emotional state) of a single agent (as opposed to an interacting community), and frequently takes advantage of explicit communications between agents (as in managing conversational protocols). Many realistic problems deviate from these conditions.

- Increasing the number of agents leads to a combinatorial explosion that can swamp conventional analysis.
- Environmental dynamics can frustrate agent intentions.
- The agents often are trying to hide their intentions (and even their presence), rather than intentionally sharing information.
- An agent's emotional state may be at least as important as its rational state in determining its behavior.

Domains that exhibit these constraints can often be characterized as adversarial, and include military combat, competitive business tactics, and multi-player computer games.

BEE (Behavioral Evolution and Extrapolation) is a novel approach to recognizing the rational and emotional state of multiple interacting agents based solely on their behavior, without recourse to intentional communications from them. It is inspired by techniques used to predict the behavior of nonlinear dynamical systems, in which a representation of the system is continually fit to its recent past behavior. For nonlinear dynamical systems, the representation is a closed-form mathematical equation. In BEE, it is a set of parameters governing the behavior of software agents representing the individuals being analyzed. The current version of BEE characterizes and predicts the behavior of agents representing soldiers engaged in urban combat [8].

Section 2 reviews relevant previous work. Section 3 describes the architecture of BEE. Section 4 reports results from experiments with the system. Section 5 concludes. Further details that cannot be included here for the sake of space are available in an on-line technical report [16].

2. PREVIOUS WORK

BEE bears comparison with previous research in AI (plan recognition), Hidden Markov Models, and nonlinear dynamics systems (trajectory prediction).

2.1 Plan Recognition in AI

Agent theory commonly describes an agent's cognitive state in terms of its beliefs, desires, and intentions (the so-called "BDI" model [5, 20]). An agent's beliefs are propositions about the state of the world that it considers true, based on its perceptions. Its

AAMAS'07, May 14-18, 2007, Honolulu, Hawai'i.

978-81-904262-7-5 (RPS) ©2007 IFAAMAS

desires are propositions about the world that it would like to be true. Desires are not necessarily consistent with one another: an agent might desire both to be rich and not to work at the same time. An agent's intentions, or goals, are a subset of its desires that it has selected, based on its beliefs, to guide its future actions. Unlike desires, goals must be consistent with one another (or at least believed to be consistent by the agent).

An agent's goals guide its actions. Thus one ought to be able to learn something about an agent's goals by observing its past actions, and knowledge of the agent's goals in turn enables conclusions about what the agent may do in the future.

This process of reasoning from an agent's actions to its goals is known as "plan recognition" or "plan inference." This body of work (surveyed recently at [3]) is rich and varied. It covers both single-agent and multi-agent (e.g., robot soccer team) plans, intentional vs. non-intentional actions, speech vs. non-speech behavior, adversarial vs. cooperative intent, complete vs. incomplete world knowledge, and correct vs. faulty plans, among other dimensions.

Plan recognition is seldom pursued for its own sake. It usually supports a higher-level function. For example, in human-computer interfaces, recognizing a user's plan can enable the system to provide more appropriate information and options for user action. In a tutoring system, inferring the student's plan is a first step to identifying buggy plans and providing appropriate remediation. In many cases, the higher-level function is predicting likely future actions by the entity whose plan is being inferred.

We focus on plan recognition in support of prediction. An agent's plan is a necessary input to a prediction of its future behavior, but hardly a sufficient one. At least two other influences, one internal and one external, need to be taken into account.

The external influence is the dynamics of the environment, which may include other agents. The dynamics of the real world impose significant constraints.

- The environment may interfere with the desires of the agent [4, 10].
- Most interactions among agents, and between agents and the world, are nonlinear. When iterated, these can generate chaos (extreme sensitivity to initial conditions).

A rational analysis of an agent's goals may enable us to predict what it will attempt, but any nontrivial plan with several steps will depend sensitively at each step to the reaction of the environment, and our prediction must take this reaction into account as well. Actual simulation of futures is one way (the only one we know now) to deal with the impact of environmental dynamics on an agent's actions.

Human agents are also subject to an internal influence. The agent's emotional state can modulate its decision process and its focus of attention (and thus its perception of the environment). In extreme cases, emotion can lead an agent to choose actions that from the standpoint of a logical analysis may appear irrational.

Current work on plan recognition for prediction focuses on the rational plan, and does not take into account either external environmental influences or internal emotional biases. BEE integrates all three elements into its predictions.

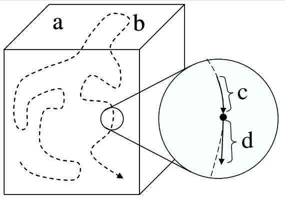

Figure 1: Tracking a nonlinear dynamical system. a = system state space; b = system trajectory over time; c = recent measurements of system state; d = short-range prediction.

2.2 Hidden Markov Models

BEE is superficially similar to Hidden Markov Models (HMM's [19]). In both cases, the agent has hidden internal state (the agent's personality) and observable state (its outward behavior), and we wish to learn the hidden state from the observable state (by evolution in BEE, by the Baum-Welch algorithm [1] in HMM's) and then predict the agent's future behavior (by extrapolation via ghosts in BEE, by the forward algorithm in HMM's).

BEE offers two important benefits over HMM's.

First, a single agent's hidden variables do not satisfy the Markov property. That is, their values at $t + 1$ depend not only on their values at t, but also on the hidden variables of other agents. One could avoid this limitation by constructing a single HMM over the joint state space of all of the agents, but this approach is combinatorially prohibitive. BEE combines the efficiency of independently modeling individual agents with the reality of taking into account interactions among them.

Second, Markov models assume that transition probabilities are stationary. This assumption is unrealistic in dynamic situations. BEE's evolutionary process continually updates the agents' personalities based on actual observations, and thus automatically accounts for changes in the agents' personalities.

2.3 Real-Time Nonlinear Systems Fitting

Many systems of interest can be described by a vector of real numbers that changes as a function of time. The dimensions of the vector define the system's state space. One typically analyzes such systems as vector differential equations, e.g., $\frac{d\vec{x}}{dt} = f(\vec{x})$. When f is nonlinear, the system can be formally chaotic, and starting points arbitrarily close to one another can lead to trajectories that diverge exponentially rapidly. Long-range prediction of such a system is impossible. However, it is often useful to anticipate the system's behavior a short distance into the future. A common technique is to fit a convenient functional form for f to the system's trajectory in the recent past, then extrapolate this fit into the future (Figure 1, [7]). This process is repeated constantly, providing the user with a limited look-ahead.

This approach is robust and widely applied, but requires systems that can efficiently be described with mathematical equations. BEE extends this approach to agent behaviors, which it fits to observed behavior using a genetic algorithm.

3. ARCHITECTURE

BEE predicts the future by observing the emergent behavior of agents representing the entities of interest in a fine-grained agent simulation. Key elements of the BEE architecture include the model of an individual agent, the pheromone infrastructure through which agents interact, the information sources that guide them, and the overall evolutionary cycle that they execute.

3.1 Agent Model

The agents in BEE are inspired by two bodies of work: our previous work on fine-grained agents that coordinate their actions through digital pheromones in a shared environment [2, 13, 17, 18, 21], and the success of previous agent-based combat modeling.

Digital pheromones are scalar variables that agents deposit and sense at their current location in the environment. Agents respond to local concentrations of these variables tropistically, climbing or descending local gradients. Their movements change the deposit patterns. This feedback loop, together with processes of evaporation and propagation in the environment, support complex patterns of interaction and coordination among the agents [15]. Table 1 shows the BEE's current pheromone flavors. For example, a living member of the adversary emits a RED-ALIVE pheromone, while roads emit a MOBILITY pheromone.

Our soldier agents are inspired by EINSTein and MANA. EINSTein [6] represents an agent as a set of six weights, each in [-1, 1], describing the agent's response to six kinds of information. Four of these describe the number of alive friendly, alive enemy, injured friendly, and injured enemy troops within the agent's sensor range. The other two weights relate to the agent's distance to its own flag and that of the adversary, representing objectives that it seeks to protect and attack, respectively. A positive weight indicates attraction to the entity described by the weight, while a negative weight indicates repulsion.

MANA [9] extends the concepts in EINSTein. Friendly and enemy flags are replaced by the waypoints pursued by each side. MANA includes low, medium, and high threat enemies. In addition, it defines a set of triggers (e.g., reaching a waypoint, being shot at, making contact with the enemy, being injured) that shift the agent from one personality vector to another. A default state defines the personality vector when no trigger state is active.

The personality vectors in MANA and EINSTein reflect both rational and emotive aspects of decision-making. The notion of being attracted or repelled by friendly or adversarial forces in various states of health is an important component of what we informally think of as emotion (e.g., fear, compassion, aggression), and the use of the term "personality" in both EINSTein and MANA suggests that the system designers are thinking anthropomorphically, though they do not use "emotion" to describe the effect they are trying to achieve. The notion of waypoints to which an agent is attracted reflects goal-oriented rationality.

BEE uses an integrated rational-emotive personality model.

A BEE agent's rationality is a vector of seven desires, which are values in [-1, +1]: ProtectRed (the adversary), ProtectBlue (friendly forces), ProtectGreen (civilians), ProtectKeySites, AvoidCombat, AvoidDetection, and Survive. Negative values reverse the sense suggested by the label. For example, a negative value of ProtectRed indicates a desire to harm Red, and an agent with a high positive desire to ProtectRed will be attracted to RED-ALIVE, RED-CASUALTY, and MOBILITY pheromone, and will move at maximum speed.

Table 1. Pheromone flavors in BEE

Pheromone Flavor	Description
RedAlive	Emitted by a living or dead entity of the appropriate group (Red = enemy, Blue = friendly, Green = neutral)
RedCasualty	
BlueAlive	
BlueCasualty	
GreenAlive	
GreenCasualty	
WeaponsFire	Emitted by a firing weapon
KeySite	Emitted by a site of particular importance to Red
Cover	Emitted by locations that afford cover from fire
Mobility	Emitted by roads and other structures that enhance agent mobility
RedThreat	Determined by external process (see Section 3.3)
BlueThreat	

The emotive component of a BEE's personality is based on the Ortony-Clore-Collins (OCC) framework [11], and is described in detail elsewhere [12]. OCC define emotions as "valanced reactions to agents, states, or events in the environment." This notion of reaction is captured in MANA's trigger states. An important advance in BEE's emotional model is the recognition that agents may differ in how sensitive they are to triggers. For example, threatening situations tend to stimulate the emotion of fear, but a given level of threat will produce more fear in a new recruit than in a seasoned veteran. Thus our model includes not only Emotions, but Dispositions. Each Emotion has a corresponding Disposition. Dispositions are relatively stable, and considered constant over the time horizon of a run of the BEE, while Emotions vary based on the agent's disposition and the stimuli to which it is exposed.

Interviews with military domain experts identified the two most crucial emotions for combat behavior as Anger (with the corresponding disposition Irritability) and Fear (whose disposition is Cowardice). Table 2 shows which pheromones trigger which emotions. For example, RED-CASUALTY pheromone stimulates both Anger and Fear in a Red agent, but not in a Blue agent. Emotions are modeled as agent hormones (internal pheromones) that are augmented in the presence of the triggering environmental condition and evaporate over time.

A non-zero emotion modifies the agent's actions. Elevated level Anger increases movement likelihood, weapon firing likelihood, and tendency toward an exposed posture. Elevated Fear decreases these likelihoods.

Figure 2 summarizes the BEE's personality model. The left side is a straightforward BDI model (we prefer the term "goal" to "intention"). The right side is the emotive component, where an appraisal of the agent's beliefs, moderated by the disposition, leads to an emotion that in turn influences the BDI analysis.

Table 2: Interactions of pheromones and dispositions/emotions

	Dispositions/Emotions					
	Red Perspective		Blue Perspective		Green Perspective	
Pheromone	Irritability /Anger	Cowardice /Fear	Irritability /Anger	Cowardice /Fear	Irritability /Anger	Cowardice /Fear
RedAlive			X	X		
RedCasualty	X	X				
BlueAlive	X	X			X	X
BlueCasualty			X	X		
GreenCasualty	X	X			X	X
WeaponsFire	X	X	X	X	X	X
KeySites	X				X	

3.2 The BEE Cycle

BEE's major innovation is extending the nonlinear systems technique of Section 2.2 to agent behaviors. This section describes this process at a high level, then details the multi-page pheromone infrastructure that implements it.

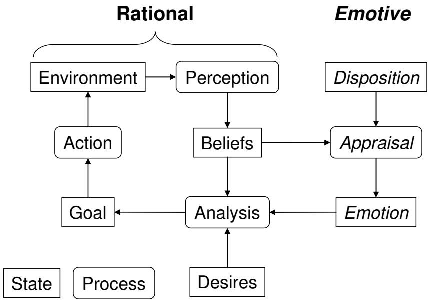

Figure 2: BEE's Integrated Rational and Emotive Personality Model

3.2.1 Overview

Figure 3 is an overview of Behavior Evolution and Extrapolation. Each active entity in the battlespace has an persistent avatar that continuously generates a stream of ghost agents representing itself. We call the combined modeling entity consisting of avatar and ghosts a polyagent [14].

Ghosts live on a timeline indexed by τ that begins in the past and runs into the future. 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 τ - t = -30, meaning that ghosts are inserted into a page representing the state of the world 30 minutes ago. At the insertion horizon, each ghost's behavioral parameters (desires and dispositions) are sampled from distributions to explore alternative personalities of the entity it represents.

Each page between the insertion horizon and τ = t ("now") records the historical state of the world at the point in the past to which it corresponds. 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 τ = t (actual time). At this point, each ghost is evaluated based on its location compared with the actual location of its corresponding real-world entity.

The fittest ghosts have three functions.

1. The personality of each entity's fittest ghost is reported to the rest of the system as the likely personality of that entity. This information enables us to characterize individual warriors as unusually cowardly or brave.
2. The fittest ghosts breed genetically and their offspring return to the insertion horizon to continue the fitting process.
3. The fittest ghosts for each entity form the basis for a population of ghosts that 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. Analysis of the behaviors of these different possible futures yields predictions.

Thus BEE has three distinct notions of time, all of which may be distinct from real-world time.

1. Domain time *t* is the current time in the domain being modeled. If BEE is applied to a real-world situation, this time is the same as real-world time. In our experiments, we apply BEE to a simulated battle, and domain time is the time stamp published by the simulator. During actual runs, the simulator is often paused, so domain time runs slower than real time. When we replay logs from simulation runs, we can speed them up so that domain time runs faster than real time.
2. BEE time τ for a page records the domain time corresponding to the state of the world represented on that page, and is offset from the current domain time.
3. Shift time is incremented every time the ghosts move from one page to the next. The relation between shift time and real time depends on the processing resources available.

3.2.2 Pheromone Infrastructure

BEE must operate very rapidly, to keep pace with the ongoing battle. Thus we use simple agents coordinated using pheromone mechanisms. We have described the basic dynamics of our pheromone infrastructure elsewhere [2]. This infrastructure runs on the nodes of a graph-structured environment (in the case of BEE, a rectangular lattice). Each node maintains a scalar value for each flavor of pheromone, and provides three functions:

- It *aggregates* deposits from individual agents, fusing information across multiple agents and through time.
- It *evaporates* pheromones over time, providing an innovative alternative to traditional truth maintenance. Traditionally, knowledge bases remember everything they are told unless they have a reason to forget. Pheromone-based systems immediately begin to forget everything they learn, unless it is continually reinforced. Thus inconsistencies automatically remove themselves within a known period.
- It *diffuses* pheromones to nearby places, disseminating information for access by nearby agents.

The distribution of each pheromone flavor over the environment forms a field that represents some aspect of the state of the world at an instant in time. Each page of the timeline is a complete pheromone field for the world at the BEE time τ represented by that page. The behavior of the pheromones on each page depends on whether the page represents the past or the future.

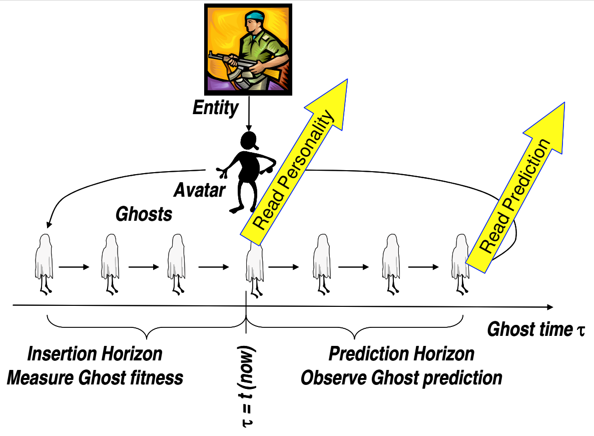

Figure 3: Behavioral Emulation and Extrapolation. Each avatar generates a stream of ghosts that sample the personality space of its entity. They evolve against the entity's recent observed behavior, and the fittest ghosts run into the future to generate predictions.

In pages representing the future ($\tau > t$), the usual pheromone mechanisms apply. Ghosts deposit pheromone each time they move to a new page, and pheromones evaporate and propagate from one page to the next.

In pages representing the past ($\tau \leq t$), we have an observed state of the real world. This has two consequences for pheromone management. First, we can generate the pheromone fields directly from the observed locations of individual entities, so there is no need for the ghosts to make deposits. Second, we can adjust the pheromone intensities based on the changed locations of entities from page to page, so we do not need to evaporate or propagate the pheromones. Both of these simplifications reflect the fact that in our current system, we have complete knowledge of the past. When we introduce noise and uncertainty, we will probably need to introduce dynamic pheromones in the past as well as the future.

Execution of the pheromone infrastructure proceeds on two time scales, running in separate threads.

The first thread updates the book of pages each time the domain time advances past the next page boundary. At each step,

- The former "now + 1"page is replaced with a new current page, whose pheromones correspond to the locations and strengths of observed units;
- An empty page is added at the prediction horizon;
- The oldest page is discarded, since it has passed the insertion horizon.

The second thread moves the ghosts from one page to the next, as fast as the processor allows. At each step,

- Ghosts reaching the $\tau = t$ page are evaluated for fitness and removed or evolved;
- New ghosts from the avatars and from the evolutionary process are inserted at the insertion horizon;
- A population of ghosts based on the fittest ghosts are inserted at $\tau = t$ to run into the future;
- Ghosts that have moved beyond the prediction horizon are removed;
- All ghosts plan their next actions based on the pheromone field in the pages they currently occupy;
- The system computes the next state of each page, including executing the actions elected by the ghosts, and (in future pages) evaporating pheromones and recording new deposits from the recently arrived ghosts.

Ghost movement based on pheromone gradients is a simple process, so this system can support realistic agent populations without excessive computer load. In our current system, each avatar generates eight ghosts per shift. Since there are about 50 entities in the battlespace (about 20 units each of Red and Blue and about 5 of Green), we must support about 400 ghosts per page, or about 24000 over the entire book.

How fast a processor do we need? Let p be the real-time duration of a page in seconds. If each page represents 60 seconds of domain time, and we are replaying a simulation at 2x domain time, $p = 30$. Let n be the number of pages between the insertion horizon and $\tau = t$. In our current system, $n = 30$. Then a shift rate of n/p shifts per second will permit ghosts to run from the insertion horizon to the current time at least once before a new page is generated. Empirically, this level is a lower bound for reasonable performance, and easily achievable on stock WinTel platforms.

3.3 Information sources

The flexibility of the BEE's pheromone infrastructure permits the integration of numerous information sources as input to our characterizations of entity personalities and predictions of their future behavior. Our current system draws on three sources of information, but others can readily be added.

Real-world observations.—Observations from the real world are encoded into the pheromone field each increment of BEE time, as a new "current page" is generated. Table 1 identifies the entities that generate each flavor of pheromone.

Statistical estimates of threat regions.—Statistical techniques1 estimate the level of threat to each force (Red or Blue), based on the topology of the battlefield and the known disposition of forces. For example, a broad open area with no cover is threatening, especially if the opposite force occupies its margins. The results of this process are posted to the pheromone pages as "RedThreat" pheromone (representing a threat to red) and "BlueThreat" pheromone (representing a threat to Blue).

AI-based plan recognition.—While plan recognition is not sufficient for effective prediction, it is a valuable input. We dynamically configure a Bayes net based on heuristics to identify the likely goals that each entity may hold.2 The destinations of these goals function as "virtual pheromones." Ghosts include their distance to such points in their action decisions, achieving the result of gradient following without the computational expense of maintaining a pheromone field.

4. EXPERIMENTAL RESULTS

We have tested BEE in a series of experiments in which human wargamers make decisions that are played out in a battlefield simulator. The commander for each side (Red and Blue) has at his disposal a team of pucksters, human operators who set waypoints for individual units in the simulator. Each puckster is responsible for four to six units. The simulator moves the units, determines firing actions, and resolves the outcome of conflicts. It is important to emphasize that this simulator is simply a surrogate for a sensor feed from a real-world battlefield

4.1 Fitting Dispositions

To test our ability to fit personalities based on behavior, one Red puckster responsible for four units is designated the "emotional" puckster. He selects two of his units to be cowardly ("chickens") and two to be irritable ("Rambos"). He does not disclose this assignment during the run. He moves each unit according to the commander's orders until the unit encounters circumstances that would trigger the emotion associated with the unit's disposition. Then he manipulates chickens as though they are fearful (avoiding combat and moving away from Blue), and moves Rambos into combat as quickly as possible. Our software receives position reports on all units, every twenty seconds.

[1] This process, known as SAD (Statistical Anomaly Detection), is developed by our colleagues Rafael Alonso, Hua Li, and John Asmuth at Sarnoff Corporation. Alonso and Li are now at SET Corporation.

[2] This process, known as KIP (Knowledge-based Intention Projection), is developed by our colleagues Paul Nielsen, Jacob Crossman, and Rich Frederiksen at Soar Technology.

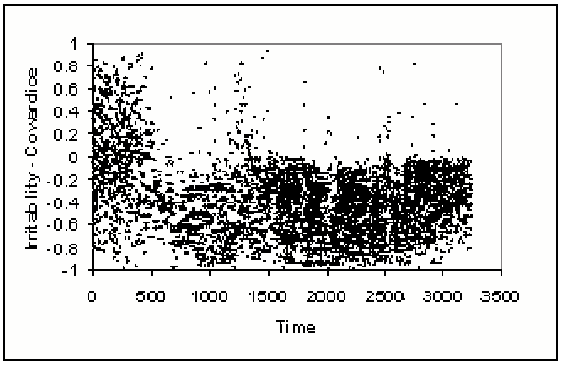

Figure 4: Delta Disposition for a Chicken's Ghosts.

The difference between the two disposition values (Irritability - Cowardice) of the fittest ghosts proves a better indicator of the emotional state of the corresponding entity than either value by itself. Figure 4 shows the delta disposition for each of the eight fittest ghosts at each time step, plotted against the time in seconds, for a unit played as a chicken. The values clearly trend negative. Figure 5 shows a similar plot for a Rambo. Rambos tend to die early, and often do not give their ghosts enough time to evolve a clear picture of their personality, but in this case the positive Delta Disposition is evident before the unit's demise.

Table 1: Experimental Results on Fitting Disposition (16 runs)

	Called Correctly	Called Incorrectly	Not Called
Chickens	68%	5%	27%
Rambos	5%	0%	95%

To characterize a unit's personality, we maintain a 800-second exponentially weighted moving average of the Delta Disposition, and declare the unit to be a chicken or Rambo if this value passes a negative or positive threshold, respectively. Currently, this threshold is set at 0.25. We are exploring additional filters. For example, a rapid rate of increase enhances the likelihood of calling a Rambo; units that seek to avoid detection and avoid combat are more readily called chicken.

Table 1 shows the detection results for emotional units in a recent series of experiments. We never called a Rambo a chicken. In the one case where we called a chicken a Rambo, logs show that in fact the unit was being played aggressively, rushing toward oncoming Blue forces. The brave die young, so we almost never detect units played intentionally as Rambos.

Figure 6 shows a comparison on a separate series of experiments of our emotion detector compared with humans. Two cowards were played in each of eleven games. Human observers in each game were able to detect a total of 13 of the cowards. BEE was able to detect cowards (= chickens) much earlier than the human, while missing only one chicken that the humans detected.

In addition to these results on units intentionally played as emotional, BEE sometimes detects other units as cowardly or brave. Analysis of these units shows that these characterizations were appropriate: units that flee in the face of enemy forces or weapons fire are detected as chickens, while those that stand their ground or rush the adversary are denominated as Rambos.

4.2 Integrated Predictions

Each ghost that runs into the future generates a possible path that its unit might follow. The paths in the resulting set over all

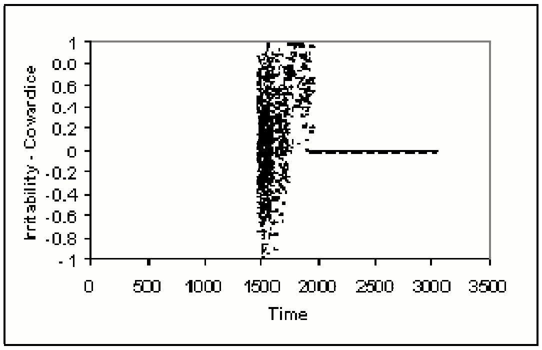

Figure 5: Delta Disposition for a Rambo.

ghosts vary in how likely they are, the risk they pose to their own or the opposite side, and so forth. In the experiments reported here, we select the future whose ghost receives the most guidance from pheromones in the environment at each step along the way. In this sense, it is the most likely future. In these experiments, we receive position reports only on units that have actually come within visual range of Blue units, or on average fewer than half of the live Red units at any time.

We evaluate predictions spatially, comparing an entity's actual location with the location predicted for it 15 minutes earlier. We compare BEE with two baselines: a game-theoretic predictor based on linguistic geometry [22], and estimates by military officers. In both cases, we use a CEP ("circular error probable") measure of accuracy, the radius of the circle that one would have to draw around each prediction to capture 50% of the actual unit locations. The higher the CEP measure, the worse the accuracy.

Figure 7 compares our accuracy with that of the game-theoretic predictor. Each point gives the median CEP measure over all predictions in a single run. Points above the diagonal favor BEE, while points below the line favor the game-theoretic predictor. In all but two missions, BEE is more accurate. In one mission, the two systems are comparable, while in one, the game-

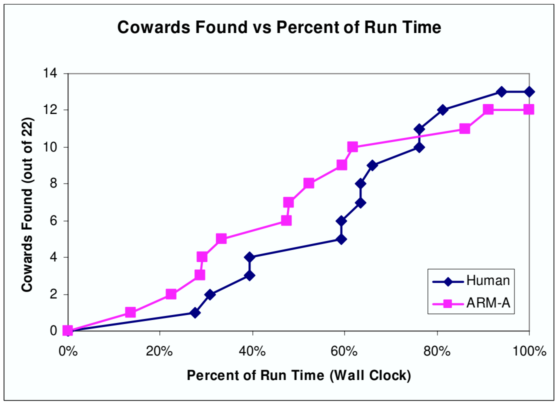

Figure 6: BEE vs. Human.

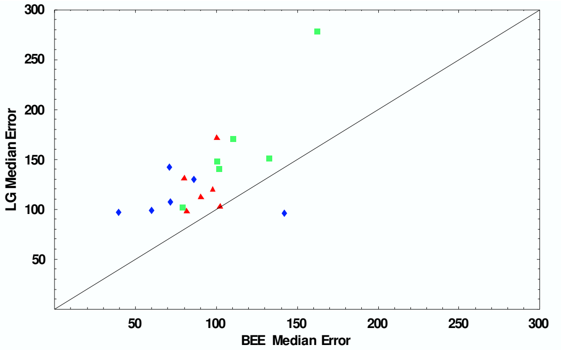

Figure 7: Median errors for BEE vs. Linguistic Geometry on each run.—Squares are Defend missions, triangles are Move missions, diamonds are Attack missions.

theoretic predictor is more accurate.

In 18 RAID runs, BEE generated 1405 predictions at each of two time horizons (0 and 15 minutes), while in 18 non-RAID runs, staff generated 102 predictions. Figure. 8 shows a box-and-whisker plot of the CEP measures, in meters, of these predictions. The box covers the inter-quartile range with a line at the median, whiskers extend to the most distant data points within 1.5 of the interquartile range from the edge of the box, squares show outliers within 3 interquartile ranges, and stars show more distant outliers. BEE's median score even at 15 minutes is lower than either Staff median. The Wilcoxon test shows that the difference between the H15 scores is significant at the 99.76% level, while that between the H0 scores is significant at more than 99.999%.

5. CONCLUSIONS

In many domains, it is important to reason from an entity's observed behavior to an estimate of its internal state, and then to extrapolate that estimate to predict the entity's future behavior. BEE performs this task using a faster-than-real-time simulation of swarming agents, coordinated through digital pheromones. This simulation integrates knowledge of threat regions, a cognitive analysis of the agent's beliefs, desires, and intentions, a model of the agent's emotional disposition and state, and the dynamics of interactions with the environment. By evolving agents in this rich environment, we can fit their internal state to their observed behavior. In realistic wargames, the system successfully detects deliberately played emotions and makes reasonable predictions about the entities' future behaviors.

BEE can only model internal state variables that impact the agent's external behavior. It cannot fit variables that the agent does not manifest externally, since the basis for the evolutionary cycle is a comparison of the outward behavior of the simulated agent with that of the real entity. This limitation is serious if our purpose is to understand the entity's internal state for its own sake. If our purpose of fitting agents is to predict their subsequent behavior, the limitation is much less serious. State variables that do not impact behavior, while invisible to a behavior-based analysis, are irrelevant to a behavioral prediction.

The BEE architecture lends itself to extension in several promising directions.

- The various inputs being integrated by the BEE are only an example of the kinds of information that can be handled. The basic principle of using a dynamical simulation to integrate a wide range of influences can be extended to other inputs as well, requiring much less additional engineering than other more traditional ways of reasoning about how different knowledge sources come together in impacting an agent's behavior.
- With such a change in inputs, BEE could be applied more widely than its current domain of adversarial reasoning in urban warfare. Potential applications of interest include computer games, business strategy, and sensor fusion.
- Our initial limited repertoire of emotions is a small subset of those that have been distinguished by psychologists, and that might be useful for understanding and projecting behavior. We expect to extend the set of emotions and supporting dispositions that BEE can detect.
- The mapping between an agent's psychological (cognitive and emotional) state and its outward behavior is not one-to-one. Several different internal states might be consistent with a given observed behavior under one set of environmental conditions, but might yield distinct behaviors under other conditions. If the environment in the recent past is one that confounds such distinct internal states, we will be unable to distinguish them. As long as the environment stays in this state, our predictions will be accurate, whichever of the internal states we assign to the agent. If the environment then shifts to one under which the different internal states lead to different behaviors, using the previously chosen internal state will yield inaccurate predictions. One way to address these concerns is to probe the real world, perturbing it in ways that would stimulate distinct behaviors from entities whose psychological state is otherwise indistinguishable. Such probing is an important intelligence technique. BEE's faster-than-real-time simulation may enable us to identify appropriate probing actions, greatly increasing the effectiveness of intelligence efforts.

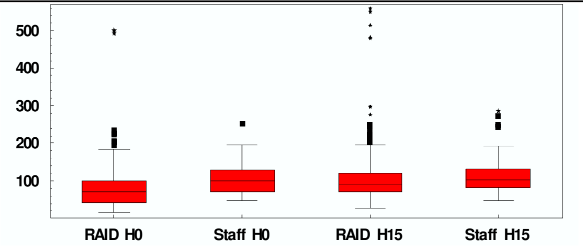

Figure. 8: Box-and-whisker plots of RAID and Staff predictions at 0 and 15 minutes Horizons. Y-axis is CEP radius in meters; lower values indicate greater accuracy.

6. ACKNOWLEDGEMENTS

This material is based in part upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. 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).

7. REFERENCES

[1] Baum, L. E., Petrie, T., Soules, G., and Weiss, N. A maximization technique occurring in the statistical analysis of prob-

abilistic functions of Markov chains. *Ann. Math. Statist.*, *41*, 1: 1970, 164-171.

[2] Brueckner, S. *Return from the Ant: Synthetic Ecosystems for Manufacturing Control*. Thesis at Humboldt University Berlin, Department of Computer Science, 2000.

[3] Carberry, S. Techniques for Plan Recognition. *User Modeling and User-Adapted Interaction*, *11*, 1-2: 2001, 31-48.

[4] Ferber, J. and Müller, J.-P. Influences and Reactions: a Model of Situated Multiagent Systems. In *Proceedings of Second International Conference on Multi-Agent Systems (ICMAS-96)*, AAAI, 1996, 72-79.

[5] Haddadi, A. and Sundermeyer, K. Belief-Desire-Intention Agent Architectures. In G. M. P. O'Hare and N. R. Jennings, Editors, *Foundations of Distributed Artificial Intelligence*, John Wiley, New York, NY, 1996, 169-185.

[6] Ilachinski, A. *Artificial War: Multiagent-based Simulation of Combat*. Singapore, World Scientific, 2004.

[7] Kantz, H. and Schreiber, T. *Nonlinear Time Series Analysis*. Cambridge, UK, Cambridge University Press, 1997.

[8] Kott, A. Real-Time Adversarial Intelligence & Decision Making (RAID). vol. 2005, DARPA, Arlington, VA, 2004. Web Site.

[9] Lauren, M. K. and Stephen, R. T. Map-Aware Non-uniform Automata (MANA)—A New Zealand Approach to Scenario Modelling. *Journal of Battlefield Technology*, *5*, 1 (March): 2002, 27ff.

[10] Michel, F. *Formalisme, méthodologie et outils pour la modélisation et la simulation de systèmes multi-agents*. Thesis at Université des Sciences et Techniques du Languedoc, Department of Informatique, 2004.

[11] Ortony, A., Clore, G. L., and Collins, A. *The cognitive structure of emotions*. Cambridge, UK, Cambridge University Press, 1988.

[12] Parunak, H. V. D., Bisson, R., Brueckner, S., Matthews, R., and Sauter, J. Representing Dispositions and Emotions in Simulated Combat. In *Proceedings of Workshop on Defence Applications of Multi-Agent Systems (DAMAS05, at AAMAS05)*, Springer, 2005, 51-65.

[13] Parunak, H. V. D. and Brueckner, S. Ant-Like Missionaries and Cannibals: Synthetic Pheromones for Distributed Motion Control. In *Proceedings of Fourth International Conference on Autonomous Agents (Agents 2000)*, 2000, 467-474.

[14] Parunak, H. V. D. and Brueckner, S. Modeling Uncertain Domains with Polyagents. In *Proceedings of International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, ACM, 2006.

[15] Parunak, H. V. D., Brueckner, S., Fleischer, M., and Odell, J. A Design Taxonomy of Multi-Agent Interactions. In *Proceedings of Agent-Oriented Software Engineering IV*, Springer, 2003, 123-137.

[16] Parunak, H. V. D., Brueckner, S., Matthews, R., Sauter, J., and Brophy, S. Characterizing and Predicting Agents via Multi-Agent Evolution. Altarum Institute, Ann Arbor, MI, 2005. http://www.newvectors.net/staff/parunakv/BEE.pdf.

[17] Parunak, H. V. D., Brueckner, S., and Sauter, J. Digital Pheromones for Coordination of Unmanned Vehicles. In *Proceedings of Workshop on Environments for Multi-Agent Systems (E4MAS 2004)*, Springer, 2004, 246-263.

[18] Parunak, H. V. D., Brueckner, S. A., and Sauter, J. Digital Pheromone Mechanisms for Coordination of Unmanned Vehicles. In *Proceedings of First International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002)*, ACM, 2002, 449-450.

[19] Rabiner, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, *77*, 2: 1989, 257–286.

[20] Rao, A. S. and Georgeff, M. P. Modeling Rational Agents within a BDI Architecture. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR-91)*, Morgan Kaufman, 1991, 473-484.

[21] Sauter, J. A., Matthews, R., Parunak, H. V. D., and Brueckner, S. Evolving Adaptive Pheromone Path Planning Mechanisms. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS02)*, ACM, 2002, 434-440.

[22] Stilman, B. *Linguistic Geometry: From Search to Construction*. Boston, Kluwer, 2000.