Proceedings of the 1999 Winter Simulation Conference
P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds.

USING ADAPTIVE AGENTS IN JAVA TO SIMULATE U.S. AIR FORCE PILOT RETENTION

Martin P. Gaupp

Rated Force Policy Analysis
Air Force Personnel Operations Agency
Arlington, VA 22202, U.S.A.

Raymond R. Hill

Department of Operational Sciences
Air Force Institute of Technology
Wright-Patterson AFB OH 45433, U.S.A.

ABSTRACT

The retention of skilled pilots is a problem that plagues the United States Air Force. After spending millions of dollars on training and education, it is disheartening to see the mass exodus of experienced aviators from the Air Force that has been occurring in the past decade. Many blame the economy, others the Air Force itself, but few are able to accurately predict how or why they are all leaving. Complex adaptive systems theory might provide some insight. By modeling the system at the pilot's level, allowing each pilot to be represented as an autonomous, independent agent continually adapting to its environment and the other agents in it, an alternate model can be built; one that accounts for the interactions among the pilots, not just their interactions with their environment. PICAS (Pilot Inventory Complex Adaptive System) is just such a model. Constructed in the Java language, the PICAS model exploits the notions of complex adaptive systems theory and employs dynamic user controls to discern retention rates over a pilot career time period. Pilots 'evolve', for lack of a better word, to a greater fitness within their environment, and in the process the model user can better determine what kind of environment needs to be created and maintained in order to ensure that trained and experienced pilots are in fact retained for their services beyond their initial service commitments.

1 INTRODUCTION

Life is complex. In model building, we eliminate the complex to simplify the problem and promote understanding. In reality, however, we sacrifice reality for simplicity. We have models that work, but we risk over abstracting reality and possibly answering the wrong questions.

According to Casti (1994, pg. 4), models are nothing but an applied set of rules. As Aristotle points out, "The whole is more than the sum of the parts" (quoted in Casti 1994, pg.171), so "if you want to study the behavior of a system composed of many parts, breaking it apart into its component pieces and studying the pieces separately won't always help you in understanding the whole" (Casti 1994, pg. 172).

This is where complex adaptive systems (CAS) theory may help. Instead of modeling top-down—the current paradigm—CAS implements bottom-up modeling. Agents are created in the model and their interactions with each other and their environment drives the system's behavior. Observing the trends in the behavior can help describe past behavior and lend insight into predicting future behavior.

The goal of the PICAS (Pilot Inventory Complex Adaptive System) model is to create artificial life to mimic the actions of real-life Air Force pilots. The computer's pilots are subjected to environmental conditions in the hope that agent reactions will provide some indication of the behaviors exhibited by real pilots in similar situations. Controlling conditions and observing behaviors is easier in such a model than doing so in real-life.

2 COMPLEX ADAPTIVE SYSTEMS

To break away from the paradigm of top-down modeling, we exploit complexity and its applications in complex adaptive systems theory. Instead of the top-down approach of contemporary simulation and modeling, complex adaptive systems theory is based on model building from the bottom up. The important paradigm shift is that complexity is really a property of the interactions between the parts of a system rather than an intrinsic aspect of the whole— complex behavior emerges from these systems.

> A small number of rules or laws can generate systems of surprising complexity. Moreover, this complexity is not just the complexity of random patterns. Recognizable features exist… In addition, the systems are animated—dynamic; they change over time. Though the laws are invariant, the things they govern change… The rules or laws generate the complexity, and the ever-changing flux of patterns that follows leads to perpetual novelty and emergence. (Holland 1995, pg. 3-4)

The Santa Fe Institute defined the characteristics of a complex adaptive system. First, complex processes generate counterintuitive, seemingly acausal behavior. Complex systems involve lots of interactions among the entities in the system and feedback plays a big role in the system's overall behavior. Complex systems also exhibit a marked diffusion of real authority. Complex adaptive systems possess a decentralized power structure spreading authority and decision making ability over a number of units (possibly all the units) in the system. System behavior is the combination of the individual behaviors of the agents acting together. Lastly, complex adaptive systems are inherently irreducible.

New models exploiting the tenets of complex adaptive systems theory have reduced predictive power. Complex adaptive systems rarely (if ever) follow the same course of action between simulations. This, however, is a realistic property. Life is not predictable. Users of complex adaptive systems are convinced their models might help describe nature, something other models struggle with. Although they may not be able to predict the future state of a system, they might be able to describe, and re-create, its past. This alone would be a great accomplishment, for to date not many models had been successful at explaining anything but the most simple of systems.

Complex adaptive system theory also helped develop the notion of artificial life.

> The ultimate goal of the study of artificial life would be to create 'life' in some other medium, ideally a virtual medium where the essence of life has been abstracted from the details of its implementation in any particular model. We would like to build models that are so life-like that they cease to become models of life and become examples of life themselves. (Langton as quoted in Levy 1992, pg. 85)

3 CONCEPTUAL MODEL DEVELOPMENT

Conceptually, PICAS models pilots as complex adaptive agents by creating a virtual environment within which the simulated pilot-agents exist and evolve. Observing the virtual environment might give us insights into actions and policies effecting the pilots' real environment.

3.1 Pilots as Complex Adaptive Agents

Our goal is to model pilot retention based on the assumption that underneath all the details lies the essence of a complex adaptive system. The pilots themselves are agents adapting to a complex environment. Pilots communicate with and influence each other's actions. They take in information from their environment, process it, and then react to it. They also take in information regarding the behaviors of other pilots (i.e., squadron-mates) and base their behaviors on this information.

The PICAS model was developed to exploit the complex adaptive systems aspects of the pilot inventory issue. By setting up a few simple rules by which the agents (pilots) in the model interact, some very complex and interesting behavior emerges. The interactions between the agents cause the emergence of complex behavior, that may very well point out ways to avoid or exploit future scenarios in the personnel management arena. By altering the computer agent's environment and agent-to-agent behavior, the users of the model may gain insights into how a real pilot might react under similar circumstances.

3.2 The Pilots' Environment

The environment within which the pilots live is represented by the playing field on the screen. In the real world, the pilots' environment is constantly changing and pilots adjust their behaviors to maximize their "happiness." A similar idea is used by PICAS. The environment within which the pilot-agents live is in a constant state of flux and agents adapt themselves to their environment.

3.3 Collection of Summary Statistics

Throughout the model's execution, the pilots are animated on the playing field, dancing about their environment and making decisions regarding their future career aspirations in the Air Force. Animation does not provide quantifiable data for analysis but does provide a means to view the emergent complex behavior exhibited.

By displaying the percentage of separations at particular times in a pilot's career, the model provides a graphical depiction of the pilots' decision making process and gives the user the ability to view the results over the course of an execution run. This information can then be used to guide policy makers and help them determine possible courses of action.

The collection of summary statistics also provides a means by which to compare different executions of the model. As with any simulation, multiple runs produce distributional information about the results. This allows the user to extract more useful and complete information from the model and use it to further refine real-world policy initiatives.

3.4 User Interaction

The ability to dynamically alter the characteristics of the agents in the model and the environment itself allows the user to conduct 'what-if' type analysis over a range of parameters. The user can test both the timing and intensity of his policy decisions and immediately see their effect on the pilots' retention and separation rates.

The parameters over which the user has control fall into two categories: those affecting the behavior of the pilot-agents and those affecting the environment. The behavior of the pilot-agents is controlled by shifting two utility curves, one representing money satisfaction, the other time-off gratification. A scroll bar determines the relative weight of each utility curve. The environmental settings, controlled by a set of scroll bars, include:

- the amount of pilot jobs available in the commercial airline market;
- the perceived pay gap between military and commercial pilots;
- the current flying operations tempo in the military.

Shifting the scroll bars attached to these settings alters the environment within which the pilot-agents live, thus causing them to alter their behavior and adapt to their new environment. Coupled with the pilot-agents individual attitudes (as measured by their utility curves), the dynamic alteration of the environmental settings allows for the emergence of complex behavior that hopefully mimics trends observed in reality. The environmental stimuli modeled are long-term in nature. Short-term policies, such as Stop Loss, are not considered in the current model.

Since PICAS is a proof-of-concept of CAS simulation for personnel analysis, some key assumptions must be clarified. We model pilot preferences using utility curves, a common technique in decision analysis (see Clemen 1996 and Kirkwood 1997). Our environmental factor choices are based on what seemed reasonable and what made model use as intuitive as possible. The scales and ranges employed on all slider bars are notional, meant merely to exercise the model.

4 SIMULATION MODEL DEVELOPMENT IN JAVA

The Java language was used to create the PICAS model in order to maximize the dissemination of the model due to Java's inherent cross platform capabilities. Added to this, the ease with which Java handles graphics allowed us to create a simple yet information rich display controlled by an intuitive graphical-user interface (GUI).

4.1 Agent Class

The agent class represents the objects manipulated in the model. Each pilot in the model is as instantiation of the agent class with characteristics particular to that pilot. The agent class contains information regarding each pilot's location in the environment (by means of a coordinate plane) as well as the parameters that govern the pilot's interactions with other pilots and the environment itself.

The agent class uses a fitness function to help determine the propensity of each pilot's intentions regarding the retention/separation issue. The fitness function is used as a means to determine which portion of the playing field attracts a given pilot-agent. The fitness function combines the environmental parameters with the pilots' attitudes to define an overall satisfaction rating for each agent. The values of the environmental parameters are used as inputs in the utility curves that define the pilots' attitudes. The hiring and paygap parameters are aggregated (via a simple average) and this value is used as the X-axis input to the pilot-agent's money utility curve. The opstempo parameter is used as the X-axis input to the pilot-agent's time utility curve. The Y-axis results from both utility curves are then combined using the ratio specified by the time/money weight scroll bar and a new fitness function is produced. This new fitness function is compared to the pilot-agent's current fitness function and used to adjust the current fitness function either up or down.

The agent class also defines exactly how the pilots interact with each other and their environment. The algorithm used here is based on Reynolds' boids simulation (see Reynolds 1998) and uses Java code from Dolan's MiniFloys Java applet (see Dolan 1998). Each pilot-agent in the simulation keeps track of a certain number of neighbors in his vicinity. Based on the agent's fitness and his proximity to other neighbors, he will alter his behavior (i.e., change his movement on the screen) to remain close to his neighbors, while at the same time following his propensity towards retention/separation. A degree of randomness is introduced into the model by 'kicking' the system a certain percentage of the time. This 'kick' results in the random shuffling of neighbors, thus allowing agents to experience effects from other agents that might not otherwise be their closest neighbor.

The playing field has three basins of attraction. The center of the playing field is a holding ground for undecided agents. The upper right-hand corner represents retention, and pilots with a propensity to stay in the service migrate to this area based on their fitness function value. Conversely, the lower left-hand corner of the playing field represents separation. Pilots whose fitness functions dip below a certain level move to this area of the playing field and eventually leave the system when they separate from the service.

4.2 PilotRetention Class

This class represents the main applet class that controls the applet's execution. The function of the PilotRetention class is to declare all the applet's variables, initialize the simulation, start the simulation, control the simulation as it runs, and then stop the simulation. These actions are all performed by various methods within the PilotRetention class.

In the initialization method, the PilotRetention class sets up the entire graphical user interface. This includes drawing the playing field, creating the environmental scroll bar controls and the pilot specific utility curves, and setting up the dynamic graph that displays the summary statistics for the user. Lastly, the initialization method instantiates the initial set of pilot-agents and gives the system its starting point.

The start method kicks off the simulation clock. The clock is implemented as the main thread of execution in the program, and uses Java's multithreading capabilities to animate the agents on the playing field by slowing down the processing speed of the algorithm at certain times. This is done by 'sleeping' the main thread of execution after each complete processing cycle through all the agents. The 'sleeping' is necessary to allow the user to see the animation on the screen. Without it, the speed at which the animation would occur would result in a blur on the screen and eliminate a great portion of the usefulness of this simulation.

After being initialized and started, the run method executes the actual applet. It cycles through the agents updating their fitness functions and processing them for movement. A small percentage of the time (currently set at 5%) it 'kicks' the system and causes the agents to randomize their neighbors. This introduces a degree of stochasticity into the model. Every 100 cycles through the algorithm, the run method introduces a new agent into the program. This ensures that there is a constant pool of agents in the system and helps to offset the departure of agents that either separate or retire.

The stop method terminates the main thread of execution. This method is never explicitly called but is activated when the program is terminated by the user.

4.3 SummaryStats Class

The SummaryStats class collects and displays quantitative information created during the execution of the simulation. This method allows the user to view model results and compare different model runs.

The statistics displayed are those regarding the separation time-frames of the agents. The percentage of agents separated at the minimum, middle, and end of career points are dynamically displayed in real time on a line graph at the bottom of the applet screen. Minimum career separations occur between 8 and 12 years-of-service. Middle career separations occur between 12 and 20 years-of-service. Finally, end of career separations occur at the 20 year point (all agents separate at this point).

The separations occur in one of two ways. If any particular agent is present in the lower right-hand corner of the screen for 100 iterations through the entire system's algorithm, he is flagged as ready for separation. As soon as this agent meets his minimum service requirements, he is allowed to separate. The other method of separation occurs whenever an agent reaches the 20 years-of-service point. At this time he is automatically separated, analogous to retirement from the service.

4.4 Graphical User Interface (GUI)

The Graphical User Interface (GUI) allows the user to both view the animation and data output of the model as it runs and dynamically alter the characteristics of the model. The actual display is divided into 5 parts (Figure 1), represented by Java's border layout manager. At the top (north position) is the title of the model. On the right (east position) are the environmental controls. On the left (west

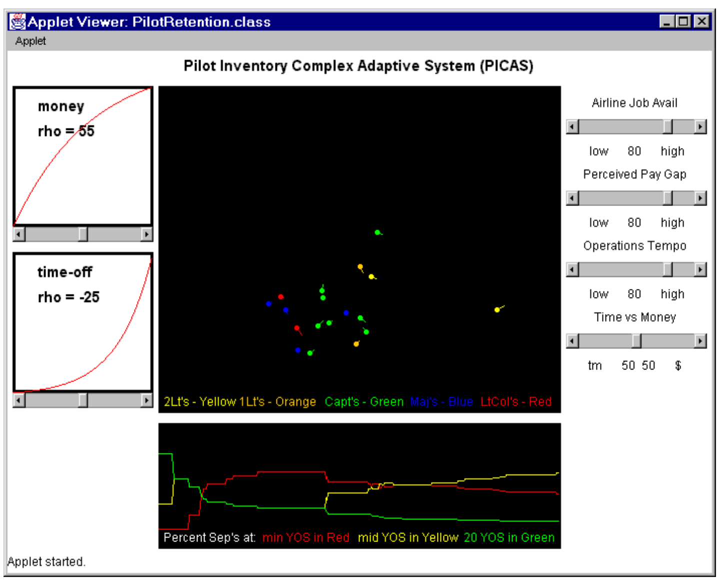

Figure 1: Screen Capture of PICAS Model

position) are the pilot controls. On the bottom (south position) is the display of the summary statistics. In the center (center position) is the pilot-agents' playing field where the animation of the pilot-agents is displayed.

The dynamic control of the model is handled by the GUI's interface with the user. Included in this interface are controls for both the environmental parameters as well as the parameters of the pilots themselves. The user is able to alter one or all of these parameters and observe the effects immediately on the simulation in progress.

The environmental parameters are controlled by means of a series of slider bars, one for each of the environmental controls included in the model. The potential for employment as a commercial pilot is controlled by the slider bar called 'Airline Job Avail'. The 'Perceived Pay Gap' slider controls the (perceived) degree of monetary discrepancy between Air Force pilots and their commercial counterparts. Lastly, the intensity of flying operations are controlled by the slider entitled 'Operations Tempo'.

The individual pilot-agent parameters are controlled by means of two utility curves, one representing money and the other representing time-off. These utility curves are controlled by a slider bar at the bottom of each curve. Sliding the bar in either direction increases or decreases the curvature of the entire curve and therefore changes the dynamics of the pilot's reaction to his environment. Utility theory is used to approximate the pilots' attitudes towards service by capturing their satisfaction with regards to the money they make and the amount of time-off they have. The Time vs. Money is the fourth slider bar, below the three environmental slider bars. This slider bar changes the weight assigned to each utility curve.

The controls in the model all work together to alter the fitness function of the pilot-agents within the system, therefore allowing them to adapt based on their personal attitudes (as captured by their utility curves) and the status of their environment (as captured by the environmental controls). If the fitness function dips below a certain threshold, the pilot-agents will migrate towards the separation attractor and eventually leave the system. Conversely, if the fitness function remains at a high enough level, the pilot-agents migrate towards the retention attractor. Undecided pilots hover near the center of the playing field.

5 ANALYSIS METHODS USING COMPLEX ADAPTIVE SYSTEMS

Models created using complex adaptive systems theory lend themselves to different forms of analysis. Their prescriptive nature allows users to glean insights about the under-winding nature of the system. However, their limited predictive power makes traditional analysis a bit harder. Exploratory modeling is an alternative analysis method.

5.1 Prescription versus Prediction

Among the biggest criticisms of complex adaptive systems theory and artificial life approaches to agent-based modeling is the fact that repeatability of experiments is very difficult. This leads to problems with developing confidence intervals for experiments, hence detracting greatly from the predictive power of such a model. Complex adaptive systems theory advocates, however, point out that complex adaptive systems are best used for their prescriptive abilities to provide insight.

Although they may lack specific prescriptive capabilities, CAS-based models allow their users to determine some aspects about the predictive power of their models. For one thing, because complex adaptive systems theory is based on chaos theory, these models generally contain certain basins of attraction that seem to cause emergent behavior to gravitate towards small zones of feasibility. Therefore, the user is able to determine with relative certainty that the end result of the model's execution will lie in some basin of attraction. If the basins of attraction are smaller than the entire feasible space that the agents can occupy, then knowledge of these basins is very informative with regards to predictive capabilities. Instead of knowing where the system is in the future, the user can tell where the system won't be in the future. This can focus attention and analysis on those areas that contain the attractors, and may lead to the discovery of scenarios and parameter settings that induce results into the preferred basins of attraction.

In the pilot retention model, this information can be very important. Looking at the playing field, there are basically three basins of attraction: the undecided basin, the retention basin, and the separation basin. During model execution, pilot-agents move towards one of these attractors. Policy decisions either induce pilots to separate from the Air Force or persuade them to stay in the service. Those decisions that result in a clear-cut decision one way or the other are easy to understand, but those that result in fairly even splits are the ones that would require further research. Again, the model user can now concentrate his efforts on exploring those options that seem to produce the ambivalent results. The model has helped focus on the important issues.

5.2 Exploratory Modeling

Exploratory modeling involves creating a database of models that can then be referenced once a given decision must be addressed. By creating models that run the entire gamut of the system's feasible space, the database created can be used to come up with a response function for all the parameters of the system. This would allow a user to determine if there are any local extrema within the system's feasible space. Such information could be used to help maximize the decisions involved in making the system run at peak performance.

For the pilot retention problem, exploratory modeling helps determine which policies yield the greatest benefit with regards to pilot retention. This allows the model user to narrow in on the models that warrant further investigation. The goal is spending less time on the modeling exercise and more time refining those policy initiatives that show promise.

6 THREE DIMENSIONAL GRAPHICAL ANALYSIS OF RESULTS

In order to provide a better graphical representation of the results presented by the PICAS model, we endeavored to create a surface plot encompassing all five parameters for each separation statistic. The problem, however, was that we needed to aggregate our five independent variables into the two dimensions required to create such a surface plot. Since the PICAS model provides two basic means for control—altering pilot attitudes or altering the system's environment—we felt this was the natural way to go.

Pilot attitudes were modeled by money and time-off utility curves that were convex (rho = 25), concave (rho = -25), or indifferent (rho = 0). The overall pilot attitude was therefore represented as the sum of the settings of all combinations of these two utility curves, resulting in an overall pilot attitude parameter with values of -50, -25, 0, 25, and 50. The overall environmental parameter was depicted in a similar manner as the sum of the three separate environmental dimensions: airline hiring practices, perceived paygap, and current operations tempo. With each individual environmental parameter set to represent a good environment (setting = 20), an average environment (setting = 50), and a bad environment (setting = 80), the resultant overall environmental parameter had settings of 60, 120, 150, 180, and 240. These two parameters, which captured various scenarios of the PICAS model, were then used as the X and Y axes on a surface plot for each set of separation data (minimum YOS point, middle career point, and end of career point).

Figure 2 is the surface plot of the separations at the minimum YOS point. As expected, an average to poor environment coupled with negative pilot attitudes results in significant separations early in a pilot's career. This is logical and provides policy makers an important piece of information. Improving either the pilot's attitudes or the environment reduces early separations. The environment seems to affect separation more than pilot attitudes. Even at an average pilot attitude separation is not an issue. But, for average environmental settings if the pilot attitude is really poor, then separation becomes an issue. Based on this information, it may be prudent for policy makers to concentrate on improving the pilots' environment rather than improving pilots' attitudes.

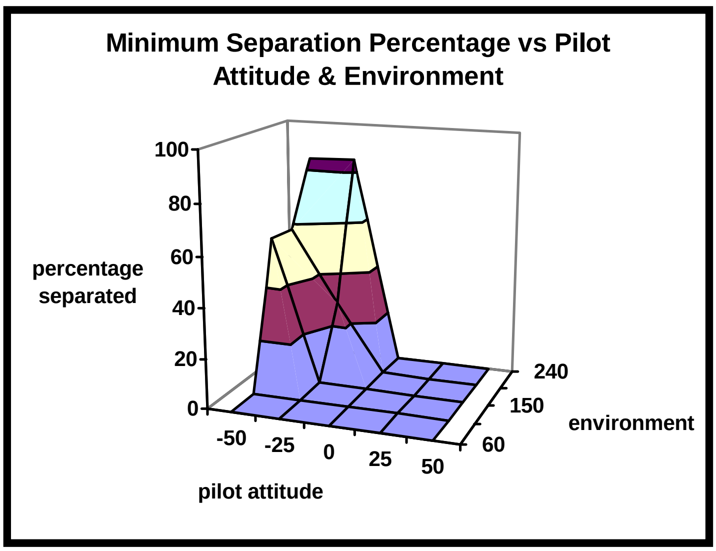

Figure 2: Surface Plot of Minimum YOS Point Separators

Figure 3 is the surface plot of the separations at the middle career point and is a bit more interesting. Separations are the highest whenever both the environment and the pilots' attitudes are poor. However, the separations at the middle of career point also increase if either one of these parameters is at its worst, regardless of the setting of the other parameter. If the environment is at its worst, then even if the pilots' attitudes are at their best, there are a significant amount of separations at the middle career point. In the same manner, if the pilots' attitudes are at their worst, and the environment is at its best, there is still a rise in the percentage of separations at the middle career point. Comparing these two extremes, however, it is apparent that the latter is not as severe as the former. A very poor environment coupled with excellent pilot attitudes results in a much greater separation percentage than very poor pilot attitudes coupled with an outstanding environment. This information is very valuable to policy makers. It indicates that given a choice between which issue to attack first, the best course of action might be to concentrate on making the environment more favorable instead of trying to improve the pilots' attitudes.

Figure 4 is the surface plot of separations at the end of a pilot's career, or those pilots that remain in the Air Force until retirement. This graph is a mirror image of the sum of the data depicted on the previous two plots. This plot shows that pilots remain primarily career minded unless both the environment and their attitudes become very poor. Retention until retirement also dips a bit if either the environment is at its poorest level or the pilots' attitudes are at their poorest level. The dip, however, is greater is the environment is poor while pilots' attitudes remain positive. Once again, this implies that decision makers should concentrate on enhancing the environment to induce retention rather than trying to alter individual attitudes.

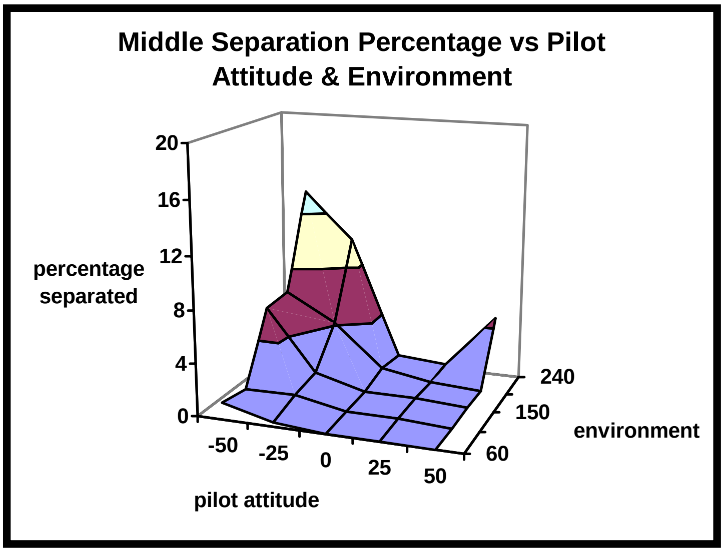

Figure 3: Surface Plot of Middle of Career Separators

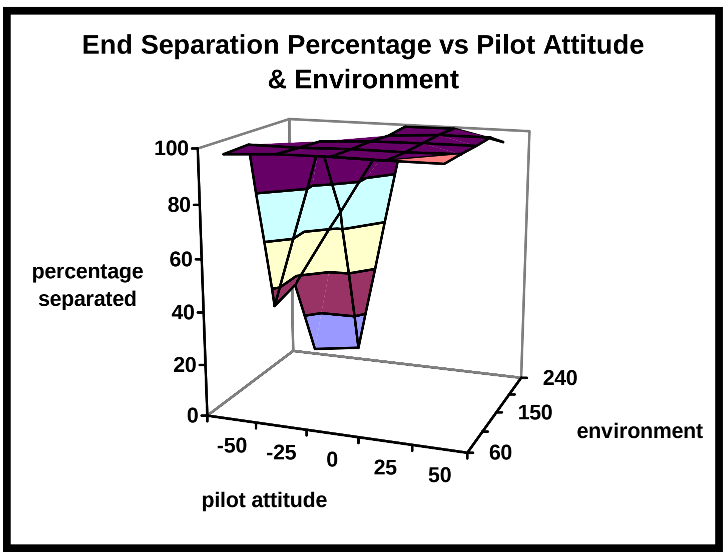

Figure 4: Surface Plot of End of Career Separators

7 CONCLUSIONS

One must not forget that computer models merely guide our decision making process. In this manner, they tend to serve two roles. In one respect, they "provide a rigorous demonstration that something is possible" (Holland 1998, pg. 241). On the other hand, they "suggest ideas about a complex situation, suggesting where to look for critical phenomena, points of control, and the like" (Holland 1998, pg. 241). A well-designed model provides insight into the system it represents. It allows the user to test out new ideas without risk of harm to the actual system. It enables the user to examine different paths of possibilities that could not be explored in the real system. A complex adaptive systems theory approach to model building, applied to the pilot inventory problem in the Air Force, introduces a new simulation and modeling paradigm to the analysis field.

PICAS is an instance of a complex adaptive system aimed at modeling pilot behavior and suggestive of a new paradigm for modeling, simulation and analysis. PICAS, as a proof-of-concept effort, incorporates some simplifying assumptions regarding pilot values and key environmental factors. PICAS is written in Java, a language whose multi-threading capabilities are well suited to agent-based modeling. Moreover, complex adaptive system models require a shift in analytical focus from prediction to prescription, and the delivery of insights versus point estimates and confidence intervals. This effort prototyped PICAS and demonstrated the analytical mindset and methodology appropriate for what may be a new generation of analytical models.

REFERENCES

Casti, John L. *Complexification – Explaining a Paradoxical World Through the Science of Surprise*. New York NY: HarperCollins, 1994.

Clemen, Robert T. *Making Hard Decisions: An Introduction to Decision Analysis, 2nd Edition*. Belmont California: Duxbury Press at Wadsworth Publishing Company, 1996.

Dolan, Ariel. "Artificial Life on the Web, Java Alife Experiments and Artist 3D Dolls". Floys and Artificial Life Web Site. World Wide Web, http://www.aridolan.com/. 20 August 1998.

Holland, John H. *Emergence – From Chaos to Order*. Reading MA: Addison-Wesley Publishing Company, Inc., 1998.

Holland, John H. Hidden Order – *How Adaptation Builds Complexity*. Reading MA: Addison-Wesley Publishing Company, Inc., 1995.

Kirkwood, Craig W. *Strategic Decision Making: Multiobjective Decision Analysis with Spreadsheets*. Belmont California: Duxbury Press at Wadsworth Publishing Company, 1997.

Levy, Steven. *Artificial Life – A Report from the Frontier Where Computers Meet Biology*. New York: Vintage Books, 1992.

Reynolds, Craig. "Boids (Flocks, Herds, and Schools: a Distributed Behavioral Model)". Boids Web Site. World Wide Web, http://hmt.com/cwr/boids.html. 20 August 1998.

AUTHOR BIOGRAPHIES

MARTIN P. GAUPP is an analyst for the Air Force Personnel Operations Agency (AFPOA) in Washington DC. He is a 1999 graduate of the Air Force Institute of Technology in Dayton, OH, where he did his thesis work on the use of complex adaptive systems theory and its application to the U.S. Air Force's pilot retention problem. For his thesis he wrote the PICAS (Pilot Inventory Complex Adaptive System) model in the Java language and used it as a 'proof-of-concept' for the application of complex adaptive systems theory to personnel related issues.

RAYMOND R. HILL is an Assistant Professor of Operations Research at the Air Force Institute of Technology. He received his Ph.D. from The Ohio State University and has held a variety of positions during his Air Force career. His research interests include simulation, statistical applications of simulation, optimization, and Java-based modeling and simulation. He is a member of INFORMS and MORS.