Assessing and Selecting AI Pilots for Tactical and Training Skill

Jared Freeman[1], Eric Watz[1], & Winston Bennett[2]
[1]Aptima, Inc.,
[2]AFRL, Airman Systems Directorate, Warfighter Readiness Research Division,
UNITED STATES

freeman@aptima.com

ABSTRACT

Pilot training is effective when subject matter experts (SMEs) fly – in exercises or simulations – as adversaries. However, SMEs are costly and scarce, thus pilots train with them infrequently. Training is potentially more accessible when Computer Generated Forces exercise pilot trainees. However, CGFs often do not respond smartly to trainees, so the training effectiveness of CGFs alone is questionable. What is needed are AI opponents with tactical expertise, agents that make tactical decisions that train, not just win. AFRL has sponsored development of agents by eight American firms. The developers fly and assess their pilot agents on a purpose-built testbed, components of which are freely available to the NATO community. In this paper, we describe the AI pilots and testbed, and present an assessment of those pilots. The AI developed here are designed to win, not to train human pilots. However, the variance in AI performance presents an opportunity to select from among these agents the ones most likely to train well. We describe several approaches to smartly selecting AI for a training task: selection by rule, by expert judgment, by model, and by empirical effects.

1.0 INTRODUCTION

Pilots learn best when they exercise against aviators who are expert in tactical flight and in training. These experts are rare. Thus, they, like the aircraft and simulators used in training, are costly and rarely available to a given trainee. Pilots can readily exercise against Computer Generated Forces in simulators and in training events that combine live, virtual, and constructive assets. However, CGFs rarely make the smart, adaptive moves of expert human opponents. Thus, their training value decreases as the expertise of the student increases, and their predictable behavior presents the potential for negative transfer of bad lessons to real engagements.

What is needed to lower cost, increase access to training, and improve the quality of training are synthetic (Red) pilots who challenge human trainees (Blue pilots) well. Creating this artificial intelligence (AI) is one mission of the Air Force Research Laboratory, Airman Systems Directorate.

The Directorate funds eight American AI firms in a program of data-driven development, execution, and assessment of tactically adaptive Red AI pilots. The program is called Not-So-Grand Challenge (NSGC), in a nod to the scale of the program, which is modest relative to its ambitions. The program has developed a testbed designed to exercise AI pilots and assess their performance. Key components of the testbed are freely available to partner nations. (Certain back-end components cannot be distributed (they are "For Official Use Only), but the testbed development team can help NATO users integrate their own replacements for those components.)

The testbed hosts six dozen systematically varied scenarios. The number and variety of scenarios supports development of robust agents. Nonetheless, we find that each pilot AI excels on some scenarios, performs poorly on others, and differs from other AI in its patterns of success. This finding has led us to ask whether and how to select among AI those that provide the best training effects, given that these agents are designed only to win simulated battles; they are not designed to exhibit instructional expertise. We plan to develop an automated librarian that picks the best agent for a given training mission. Below, we present several methods for creating this librarian, methods that vary in cost and effectiveness. The methods, in ascending order of cost, employ simple rules that leverage AI performance data, use expert judgment, apply expertise to model and automate adaptive instruction, or apply empirical data concerning training effectiveness.

2.0 AI PILOTS AND THE AGENT TESTBED

2.1 AI Pilots

The NSGC funds development and application of diverse architectures for creating AI pilots. These architectures range from production systems (rules selected to attain dynamic goals and mediated by conflict resolution mechanisms), state transition networks, mathematical models of cognition, to reinforcement learning algorithms. Here, we briefly summarize the architectures used in the NSGC. We then turn to the testbed on which these agents fly, and with which program leadership assesses them.

TiER1 Performance Solutions, LLC, applies a hybrid architecture that integrates two different human behavior representations. A task network model represents operator goals or functions in graph form. An accumulator model aggregates data over time to control transitions through the task network.

Stottler Henke Associates, Inc., uses the SimBionic architecture, which implements an agent as an integrated set of behavior transition networks. This open source architecture supports a dynamic scripting machine learning algorithm, developed to adapt the behavior of agents by learning from experience.

Aptima, Inc., employs reinforcement learning techniques to infer tactical state and appropriate tactical response. These populate a Behavior Definition Language (BDL) that expresses goals, tactical state, behavioral constraints, actions, predictive measures and other attributes necessary for intelligent agent behavior. BDL is input to Soar agents.

Soar Technology, Inc., applies the principles of the Soar cognitive architecture, a production system that dynamically processes environmental states to accomplish goals. When such a system is defined with a high level of fidelity, the resultant behavior model captures a set of policies that map inferred state with intended behaviors, a paradigm that lends itself to scalability, real-time execution, and explainability.

Eduworks Corporation employs Brahms-Lite, a streamlined and interoperable evolution of Brahms, a government-developed agent modeling framework created to design, simulate, and develop work systems composed of humans and technologies. Brahms-Lite is particularly useful for representing and simulating human-human and human-machine interaction and for working in concert with other simulations and agent architectures.

Discovery Machine, Inc., applies a cognitive architecture called DMInd that represents hierarchies of prespecified problem spaces and response strategies, which are retrieved as a function of fit to context. These functions are designed to achieve accurate inference and tactical action concerning tactics.

CHI Systems applies its Personality-enabled Architecture for Cognition (PAC), a system that uses narrative threads to control perception and behavior. PAC explicitly represents characteristics such as risk tolerance and perception of threat that may vary between individuals or national forces.

Charles River Analytics uses its reactive behavior modeling framework, Hap, to provide realistic, dynamic, and situation-adaptive behavior. Hap is designed to support highly parallelized behavior and manage the exchange between potentially competing goals (e.g., performing a Sweep mission and engaging a specific adversary). Within Hap, to address combat pilot reactions and decision-making, we have constructed expert behaviors grounded in recognition-primed decision-making, which focuses on rapid loops of information gathering, situation assessment, and decision-making.

These developers build AI pilots that are optimized to fight robustly, across a range of simulated battles. (None are designed with instructional capabilities.) The developers craft or train their agents on their choice of 72 unclassified scenarios created by expert pilot trainers. These scenarios require AI pilots (Red air) to generate and coordinate tactical actions in the face of different opponent presentations and limited resources (Blue air). The 72 scenarios offer the initial conditions of two Blue adversary aircraft (single group, azimuth, and range), the maneuvers they perform from each presentation (4 variations), and the fuel and weapons load variations for the two Red aircraft (6 initial variations of fuel, from just above Bingo to comfortably above Joker with weapons, from zero to full load per aircraft).

2.2 The AGENT Testbed

The AI pilots fly scenarios on the Agent Generation & Evaluation Networked Testbed. AGENT was developed by Aptima for AFRL to facilitate independent exercise and assessment of AI pilots by the distributed modelers with minimum support from SMEs. Here, we describe the architecture of the AGENT testbed.

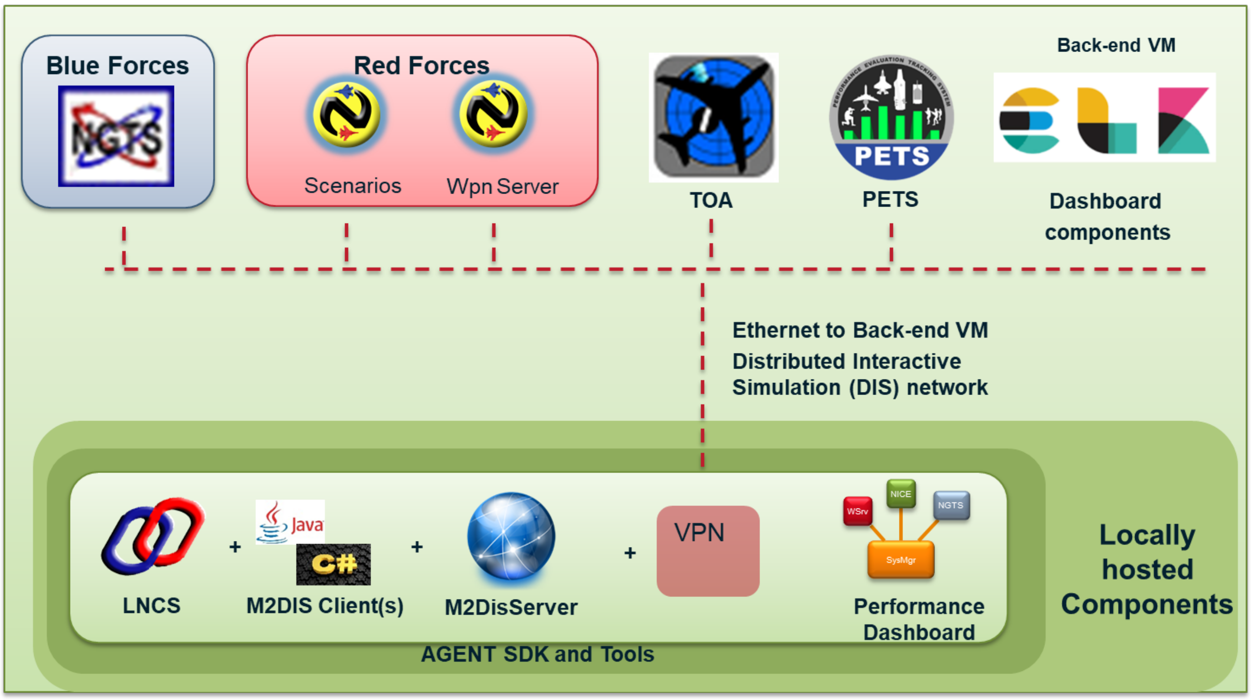

Figure 1: The architecture of the AGENT testbed.

The AGENT testbed [1] is a system of systems that includes components responsible for: generating the constructive Red and Blue players, robust data recording & playback tools, and data capture and performance measurement. AGENT is capable of integrating multiple Computer Generated Forces (CGF) systems and uses the Next Generation Threat System (NGTS) to generate Blue forces along with AFRL's Network Integrated Combat Environment (NICE) to generate Red Forces. AGENT employs a CGF-agnostic architecture that communicates AI pilots' intent to multiple distributed CGFs via a Distributed Interactive Simulation (DIS) network connection. Performance measurement and data collection within AGENT are handled by the Performance Evaluation and Tracking System (PETS™), which uses the testbed's DIS data to drive operationally-validated performance measurement and assessment algorithms and stores the resulting information in a data lake architecture.

AGENT is a distributed systems design that abstracts away the details of the underlying modeling and simulation (M&S) architecture from the AI pilot agent. AGENT provides developers with multiple levels of data access and abstraction. At the syntactic level the testbed API's provide wrappers around the native M&S data; at the semantic level the testbed provides access to the Tactical Observation Agent (TOA) computed air picture that simulates the level of information available to pilots from an airborne or ground control station; at the pragmatic level AGENT provides a high level Fighter Combat-Tactical Awareness Capability (FC-TAC) API focused on overall state of and beliefs about the environment.

The AGENT Software Development Kit (SDK) provides developers with multiple language bindings and supports agent development in Python, C# and Java. The AGENT SDK allows developers to maximize their agent's running time through a batch processing capability wherein developers queue up multiple scenarios that are then automatically run by the AGENT framework. Visualization of agent performance is done through the Live, Virtual and Constructive Network Control Suite (LNCS), which provides 2D and 3D geospatial views of the DIS data and automatically records all scenarios run on AGENT. Developers can use LNCS to replay and analyse previously executed AI agent performance.

3.0 ASSESSMENT OF PILOT AI

The NSGC hosts an annual assessment event, in which AI pilots execute one scenario (from the library of 72) that is announced in advance, and one scenario that is announced at the event. SMEs comment on each scenario execution, to provide some immediate feedback to the developers and to launch discussion between them. In a subsequent meeting, the SMEs and program leaders perform a systematic assessment and comparison of agents. This assessment combines automated measures and SME judgment to score and characterize each agent.

PETS generates measures during this event (and during the modelers' development exercises): one measure of effects that combines Blue kills and Red losses, and several process measures: fratricide, deconfliction between Red aircraft, aircraft location relative to adversary weapon lethality range, time inside adversary's weapon lethality range, and airspeed. These measures each have up to four levels of assessment: successful, passing, not optimal, and failure.

The use of automated measures frees SMEs from the tedium of counting observable behaviors and events, so that they can apply nuanced judgment and generate insightful guidance for developers.

SMEs generate one measure of effects -- a kill ratio -- and several process measures concerning the quality of intercept geometry, adherence to contract, split decision, deconfliction, spike awareness, post merge maneuver, fuel management, weapons management, element targeting/sorting. In addition SMEs, assess the overall tactical intelligence of the AI pilot, and the agent's provision of explanations of its behaviors.

Assessments over two recent years of program activity illustrate high variance in AI performance. For example (Table 1), AI pilot E scored between 13% and 100% on tactical effects (left table) across four scenarios, and the seven AI scored between 18% and 96% on tactical process (right table) in scenario #2.

Table 1: Scores for seven AI pilots (A-G) over four scenarios illustrate variance in tactical effects (kill ratios, left)) and tactical process (right) both within AI between scenarios (rows), and between AI within scenarios (columns).

	Measures of Tactical Effects					Measures of Tactical Process			
AI Pilot	Scenario #1	Scenario #2	Scenario #3	Scenario #4		Scenario #1	Scenario #2	Scenario #3	Scenario #4
A	100%	50%	50%	50%		15%	25%	30%	8%
B	25%	50%	100%			33%	18%	86%	81%
C	13%		50%	50%		16%		34%	1%
D	100%	100%	50%	50%		54%	95%	20%	22%
E	13%	50%	100%	50%		60%	33%	97%	90%
F	88%	75%	75%	100%		13%	96%	87%	90%
G			75%	75%				52%	13%

4.0 METHODS FOR SELECTING AI AGENTS TO TRAIN HUMAN PILOTS

The AI pilots vary in their performance between scenarios. Thus, no one agent is the ideal training partner across all scenarios. While it is certainly possible to select, for all trainees and training events, the one agent that performs best on the average over all scenarios, this seems likely to produce inferior training for most trainees. The variance in performance presents an opportunity to choose the most tactically competent agent agent for each scenario and, ideally, the most instructionally effective agent for each trainee who executes that scenario.

An automated librarian could accomplish this task. But, how would this librarian work? Here, we present several schemes. We order them from least to most costly, and arguably least to most reliable and instructionally effective.

Select by rule: The assessment technique described above scores AI by proficiency. An automated librarian would use these rankings to select the most tactically proficient AI pilot as the adversary for a given scenario. The challenge is to do this efficiently at scale. NSGC is technically able to run each AI through all 72 scenarios. This would enable us to automatically compute the 504 ranking scores (7 agents x 72 scenarios) of agents on tactical effects (kill ratios) and tactical process per scenario. A more robust set of rankings might be generated by running each agent through many instances of each scenario against synthetic Blue pilots that vary systematically or probabilistically in the timing or accuracy of their maneuvers.

It is, of course, possible to generate rankings of AI pilots using only or also SME judgments. However, the scale of the work might require that SMEs judge AI pilots on a sample of scenarios. This sampling could be done smartly by the SMEs themselves, or by clustering scenarios on the automated measures and selecting the central scenario in each cluster as a representative for its neighbors.

Some variants of the librarian might train better. One variant might rank all agents by their performance at critical moments (or vignettes) within each scenario (not just rank them by their performance across the complete scenario). For example, Red AI might be ranked on its decisions to pursue Blue or return to base at moments when Blue turns away and Red fuel is low or ample. These ratings would enable the librarian to swap agents within scenarios, effectively ensuring that the sharpest AI is engaged in each vignette.

Another variant of the librarian might choose an AI pilot that slightly underperforms its peers or that performs well on some process measures but poorly on others. This strategy gives the trainee experience with a range of Red behaviors in one tactical situation, and in this sense it is a substitute for systematically varying Blue behaviors across many runs of the same scenario.

Select by expert judgment: An alternative librarian leverages the training effects of the AI as predicted by training experts. SMEs would examine the actions of the most tactically proficient Red pilot agents in each scenario, as determined by the measurements discussed above. The SMEs would then predict the training effects of actions by each agent in each scenario. Critically, SMEs would make this prediction for imagined trainees at several levels of expertise. For example, a SME might predict that the tactical decisions of a specific Red AI will grow the skills of inexperienced pilots, but have no effect on experienced ones. During training, the librarian would select the AI pilot that SMEs predict will have the greatest training effect given an assessment of the trainee and the identity of the current training scenario.

A useful variant of this librarian periodically tests the SME predictions of training effects. It would substitute an alternative agent in the current scenario, measure the effects on trainee performance, and report the difference in effects obtained by different agents. These data would be used to refine SME predictions of training effects.

Select by probabilistic model: Yet another librarian applies expert judgements in a formal, computational model to select agents and scenarios that improve learning over the course of training. (In contrast, the librarians above select an agent given a scenario, and thus attempt to optimize training for that scenario only). This librarian represents trainees, agents, scenarios, and measures in a formal model that estimates the probabilistic effects of selecting a scenario and agent (i.e., taking a training action) on trainee learning, given imperfect knowledge of those effects and of the current state of trainee expertise. A computational solver is then applied to that model to produce a training policy (a large and potentially complex set of rules). The automated librarian uses this policy and measures of trainee performance to dynamically choose the next scenario and agent, the pair that will most certainly pull each trainee closest to expertise. Military research studies have demonstrated the impact of such models (e.g., a Partially Observable Markov Decision Process, or POMDP) and policies on learning [2][3].

This formal approach has several benefits. First, it automates the selection of scenarios and agents to fit each trainee's needs, and revises that selection continuously as training measurements roll in. Second, policy-driven, dynamic curricula optimize learning over the full course of training (not just the current training trial). Third, this technique systematically addresses unavoidable uncertainty concerning trainee expertise and the effects of training actions. Finally, this approach captures the empirical data needed to iteratively reduce uncertainty, refine the model, compute better training policy, and improve training outcomes.

Select by empirical effects: The formal method, above, uses expert estimates of training effects to compute an optimal selection of scenario and agent. Empirical data concerning training effects would produce better selections than would expert estimates, but at considerably higher cost in laboratory data collection. That cost may be warranted where the stakes are high, as in training pilots in novel tactics against novel threats, or where there is significant uncertainty concerning training effects.

When these special cases are identified, training experiments would be run to systematically measure the effects of each agent and scenario on pilot proficiency. These experiments would consider the effects of scenarios, agents, their combinations, and their ordering. Thus, any given experiment could be quite large, even if it merely sampled the large factorial space. How large is that space? Consider a small case of 5 scenarios x 5 AI x 3 levels of trainee proficiency x n = 8 trainees per level. Full factorial design requires 600 experimental training trials (irrespective of order effects and other confounds). Where those costs are warranted, the experimental data would feed a librarian – either rule based (above) or model-based (above) – that select a scenario and agent that demonstrably optimize training effectiveness. However, we note that the size of these experiments is probably not sufficient to train sophisticated and data-hungry machine learning algorithms (such as reinforcement learning) that might provide additional benefits in training administration.

5.0 FUTURE RESEARCH

Training AI agents requires robust and sizeable data sets. Accurately generating data in the tactical air combat domain requires high fidelity systems such as those described above, and these requirements present a ceiling to AGENT's vertical scalability, because scenarios must be executed at or close to wall-clock speed to achieve high fidelity. This limits the amount of training data that an AGENT instance produces during any given time interval. Therefore, an option to be explored is the horizontal scalability of such a testbed architecture. While a single scenario cannot execute at 100x, a testbed that scales horizontally to run 100 simultaneous scenarios will achieve the intended result of generating significantly more agent AI training data in a shorter time period.

Our primary goal concerning the assessment techniques used in the NSGC is to automate measurements that are most predictive of mission effects, and those that are more accurately and reliably computed than observed by SMEs. If successful, the automation of these measures should free SMEs to make subtler judgments and, critically, generate feedback for pilots that accelerates learning.

Finally, we framed the automated librarian as a tool that leverages the variance in performance by tactical AI to improve training effects. However, these methods would succeed even using AI that have instructional intelligence (as well as tactical skill). A set of agents that is diverse with respect to training strategies would likely show variance in its tactical effects and its trainee effects. So, smart selection between instructionally proficient agents is worthwhile. The hypothesis to test is whether smart selection among tactically proficient AI produces training outcomes that are at least as good as selection among AI that are tactically and instructionally smart but also more costly to build and maintain.

6.0 ACKNOWLEDGMENTS

This article is based upon work supported by the United States Air Force Research Laboratory, Warfighter Readiness Research Division 711 HPW/RHA, under Contract FA8650-16-C-6698. This article is cleared for public release as of 28 August 2020, 88ABW-2020-2719.

Thanks to members of the following organizations for the concepts and agents they have developed in this program: Aptima, CHI Systems, Charles River Analytics, Discovery Machine, Eduworks, Soar Technology, Stottler Henke Associates, and Tier1 Performance Solutions. Thanks to our program SMEs: Bruce "Carp" Carpenter and Donald "Harley" Simones. Finally, thanks to Joshua Ziegler for discussions concerning the work and concepts reported here.

7.0 REFERENCES

[1] Freeman, J., Watz, E., and Bennett, W. Jr. (2019). A testbed for developing & evaluating AI pilots. Proceedings of ITEC 2019. 14-16 May 2019. Stockholmassan, Sweden.

[2] Carlin, A., Kramer, D., Nucci, C., Oster, E., Freeman, J., Brawner, K. (2016). An adaptive AAR capability for GIFT. Robert Sottilare & Scott Ososky (eds.) Proceedings of the 4th Annual GIFT Users Symposium (GIFTSym4). U.S. Army Research Laboratory, Human Research & Engineering Directorate, Orlando, Florida. ISBN 978-0-9977257-0-4.

[3] Levchuk, G., Shebilske, W., and Freeman, J. (2012). A model-driven instructional strategy: The benchmarked experiential system for training (BEST). In P. J. Durlach & A. M. Lesgold (Eds.) Adaptive Technologies for Training and Education. Cambridge, UK: Cambridge University Press.