Intelligent Agent Applications in Virtual Experimentation

Grant Degenhardt  
Virtual Warfare Center

Steven W. Cooke  
Virtual Warfare Center

The Boeing Company  
mailto:grant.d.degenhardt@boeing.com

Abstract. Advances in technology have sparked the expectation of modeling the real world with much greater fidelity and accuracy. Corporate industry and the military see the need to use these “emulations of the real world” to perform predictions and analysis of alternatives much better than ever before and ultimately improve their business or service. But to attain these “emulations of the real world” many challenges in modeling and simulation will need to be overcome that have yet to be considered due to the nature of complex adaptive systems.

This paper addresses the needs and challenges inherent in creating, managing and evaluating complex adaptive systems in the context of virtual experimentation with a focus on how intelligent agents that learn and adapt are a major part of the solution. Reducing the operator-to-virtual entity ratio in virtual experimentation is a cost-driven incentive. An even tougher challenge is reducing this ratio for a complex adaptive system while avoiding simply producing a large constructive simulation. Other challenges include making the simulation more immersive and using intelligent agents to automate and manage scenario generation, simulation execution, and real-time data logging and processing.

Intelligent agents that learn and adapt are ideal and also necessary for evaluating complex adaptive systems because they can be modeled after humans who successfully traverse and navigate the real world; the largest complex adaptive system. These adaptive intelligent agents do not need to be logically complete or consistent like theorem provers that look for absolute truth. Families of agents that are each different, but all similar, will be initialized with subject matter expert (SME) knowledge. Agents will learn and adapt to the environment to optimize attainment of goals and satisfaction of other metrics. The SME knows how the agent is to behave and the SME knows how the agent is to learn because he knows when and how an agent’s behavior needs to be adjusted through iteratively viewing its behavior and interactions within the simulation environment.

1. INTRODUCTION

Intelligent agent applications are necessary when conducting virtual experimentation within a large environment on the scale of a complex adaptive system. Intelligent agents can be applied to help configure, manage and analyze the simulation as well as provide realistic entities to populate and enrich the environment. A description of several intelligent agent applications designed to aid in the simulation and evaluation of complex adaptive systems for the use in virtual experimentation are outlined in this paper.

2. SYNTHETIC ENTITIES

A key requirement for virtual experimentation is that the simulation environment be realistic and immersive. The human participants in the virtual experiment need to be able to suspend their disbelief in the simulated nature of the experiment and imagine they are immersed in an actual “real-world” exercise. This realism is difficult and expensive to satisfy in the presence of recognizable and often glaring reminders that the experiment is being conducted within a simulation environment. These glaring reminders are often caused by the behavior of the simulation-controlled virtual entities that populate the environment in large numbers.

Due to the cost of human participation in an exercise, simulation-controlled virtual entities (synthetic entities) far outnumber the human-controlled virtual entities (manned entities) in a given virtual experiment. Due to the large numbers of synthetic entities needed in an experiment and the corresponding large amount of interaction between synthetic entities and manned entities, the impact of the synthetic entity’s behavior on the realism of the environment is immense. The more the synthetic entities behave like humans the more realistic the simulation environment will seem to the human participant.

Synthetic entities need to appear reactive to their environment. Their behavior needs to appear purposeful as if there is a reason for most if not all of what they are doing. While purposeful, their behavior
Needs to also appear at graduated levels of performance ranging from flawed to exceptional as if due to factors such as the imperfection of perception, the lack of information, the result of strenuous training, the impact of emotions, etc. The synthetic entities need to appear as if they are learning from their experiences and can adapt to the environment in which they have been placed. They also need to appear similar but at the same time unique or dissimilar to other synthetic entities with the same goal or role in the scenario. Intelligent agent technology is an obvious candidate for satisfying these requirements making the behavior of the synthetic entities more human-like or more intelligent.

An intelligent agent is inserted to take the place of an actual human participant in an experiment. Fighter pilot operators, ground vehicle operators and command and control decision-makers are just a few examples of where intelligent agents can be inserted as synthetic entities.

3. INTELLIGENT AIDS

Another use for intelligent agents is in the area of intelligent aids. In these cases, the intelligent agent does not take the place of a human participant in the experiment but does provide service or aid to humans interacting with the experiment with varying levels of autonomy. Unlike the synthetic entities, intelligent aids are not limited to human levels of intelligence or cognitive ability. Human-like thinking and goal-attainment can be diminished or heightened to isolate the independent value of the rules in the nature of an experiment. Increasing or decreasing memory, computation or sensory capabilities for an experimental purpose is a powerful and extensible tool for virtual experimentation. The human-like behavior of the intelligent agents give these aids purpose and the ability to learn and adapt to their environment, role or to the desires or tendencies of the human using the aid. These intelligent aids include Automatic Scenario Generation, Real-time and Post-process Knowledge Extraction and many applications in the wide spectrum of Simulation Execution Management.

3.1 Automatic Scenario Generation

A simulation environment characterized as a complex adaptive system would contain a large number of entities, a rich and textured world including most likely high fidelity terrain, roads, bridges, buildings, urban cities, weather, etc. It takes a large amount of time for developers and analysts to generate and populate databases to define these complex worlds. This process area is ripe for the insertion of intelligent agents to take over some of the decision-making responsibilities that lead to the generation of these large and rich worlds. Where do we first insert them? How do we control their behavior so the worlds are valid or at least realistic? How much autonomy or responsibility do we give them without human supervision? These are all good questions that along with many other questions need to be answered for all intelligent agent applications in virtual experimentation.

The goal is to transfer into the intelligent agent the strategies and goals that humans use to make decisions. In the case of Automatic Scenario Generation (ASG) agents, the strategies and goals are based on the rules developers and analysts use to generate and define the simulation environment. The analyst looks to information to decide what type and density of entities, objects and terrain will be present in a given region. The analyst uses this information and the scenario tools he has to generate a simulated world. Let’s use an example to explore how ASG agents could be used to aid in the generation of synthetic worlds.

An analyst has been given the job of generating a simulated world in a 10 square mile area situated in a valley. The analyst’s mission is very similar to that of a fighter pilot’s war-time mission. They both have goals. They both have data sets or information they are pulling from and then applying strategies to in order to come to decisions. Historically more time and money has been spent on modeling and simulating the world that the pilot will traverse, navigate and manipulate. The analyst traverses a world with much different data sets but there is no reason an ASG agent could not perform the functions or tasks as long as the world was articulated to a high enough fidelity and interfaces were defined with the databases and tools needed to manipulate that world.

The level of autonomy of the ASG agents would most likely be a scalable or at least a configurable parameter. If rapid real-time generation is a requirement then highly autonomous ASG agents would be given authority to retrieve information from known sources and automatically construct the virtual world with human management by exception. Decreasing the autonomy of the system would give the operations analyst the ability to create a more specific world using ASG agents to help scan large amounts of data, detect and manipulate objects, relationships or communications of interest and methodically build a rich and textured world while being able to visualize his progress with ASG agent-controlled visualization aids.

Several different strategies could be used to generate the world in the valley of this example. Depending on requirements, one strategy the analysts could use is to base the valley world on an actual real-world place for which data is available. Road networks, urban landscapes, population densities for culture, etc. could all be pulled from a real-world database. Interfaces and data of the real-world database and simulation environment would need to be accessible and comprehensible for an ASG agent. This strategy may prove very useful and satisfy the analyst’s goals when the valley world is to be the location for a crucial mission evaluation in the virtual experiment.
Another strategy the analyst may use depending on requirements is to create a fictional valley world. All human participants in the experiment that enter the valley world and interact with its inhabitants need to only believe they are interacting with a realistic scenario and not a particular scenario. In this case, using some simple rules and stochastic data and methods, an ASG agent could “grow” a fictional world within the valley. Is the valley an industrial, commercial or residential area? Is the area large enough for 1, 2 or 3 elementary schools? How close would families typically live from a school? Where would a convenient location be for a grocery store? These questions all point to relationships between geographical locations, the objects of the landscape and the residents of that area that an agent could easily use along with region specific tendencies to strategically and stochastically generate a realistic, fictional world that could add richness and realism to a virtual experiment.

3.2 Simulation Execution Management

One of the huge challenges in performing a virtual experiment on the scale of a complex adaptive system is in managing the execution of the large and complex simulation. The challenges range from making sure there are enough computer resources to accommodate execution to knowing where and when key opportunities for analysis or demonstration will occur.

Virtual experimentation obviously includes human-controlled or manned entities interacting with the simulation environment. One effect of human interaction in the experiment is the loss of predictability. Even large constructive simulations of the size to be considered complex adaptive systems are often difficult or impossible to predict much like the behavior of the balls on a pool table immediately following a break. Add humans in the loop and any sense of consistent predictability is lost. Post-processing of logged data is and will always be used for analysis but the essence of virtual experimentation is to have real-time insight or access to what has happened or is going to happen.

Just like having manned entities in the experiment, implementing a large number of human analysts, test conductors and simulation engineers is costly and difficult to attain. The goal is to shift the roles and responsibilities of a large number of humans into intelligent agents. The intelligent agents would have varying levels of autonomy depending on their roles and the maturity of the simulation. Once again, the goals, rules and strategies used by the intelligent agents would be garnered from the human analysts, test conductors and simulation engineers; the subject matter experts. The intelligent agents would make decisions like humans but would have access to heightened computation, data input/output and memory like computers. Since many of these intelligent agents would be directly supporting a smaller number of human “supervisors” their ability to learn and adapt to their roles and the tendencies of their human supervisor become essential and a capability that could vastly increase communication, optimization, efficiency and productivity of virtual experimentation.

3.3.1. CPU Loading Optimization

Simulation engineers spend a great deal of time and effort on making sure the execution of the simulation in a virtual experiment has enough computer resources to keep the simulation properly executing in real-time. This scheduling and resource task is not trivial and usually requires human intellect to achieve. How much CPU power and memory required is dependent on many dynamic variables of the simulation. Number of foreground entities, number of background entities, number of entities in proximity of all types of sensors, number of computer processors available, the current allocation of entities to computer processors and the amount of random access memory are all factors that effect how much memory and CPU power is needed. Due to the nature of virtual experimentation these factors are all dynamic during the execution of the simulation. The ability to predict their values or state could greatly increase the ability to manage the execution of the simulation, optimize the use of computer hardware and maximize the size, complexity and productivity of the virtual experiment.

CPU Loading Optimization (CLO) agents could be used to predict the future state of synthetic and manned entities and estimate their need to be foreground or background entities. Synthetic entities in close proximity or interaction with a manned entity are defined as foreground entities and often are modeled at a higher fidelity than background entities that do not have close interaction with a manned entity and are often just used to “fill up” the environment. Close proximity or interaction is not intended to be only a geographic or spatial relationship and could just as easily be related to role, mission, identity, nationality, etc. Obviously, the need for foreground entities to morph into background entities and vice versa is a needed capability to optimize computer resources and another responsibility that could be realistically shifted to a CLO agent.

CLO agents that predict and manage the state of synthetic and manned entities could then feed their output to another CLO agent that could predict the necessary computer resources to keep the simulation executing in real-time. These CLO agents would also be aware of the state of the computer network, the current allocation of synthetic entities to computer processors and any redundancy requirements including contingency plans in case of a computer or processor going down. These agents could then manage the CPU loading of the computer network by reallocating the processing of synthetic entities to specific computer processors either autonomously or when prompted by a simulation engineer depending on the level of their autonomy and maturity of the system. The CLO agents
would use rules and strategies based on simulation engineer expertise to re-allocate the CPU loading, learn from previous decisions which allocations in what context obtained the best results and adapt their strategy to include this learned behavior.

### 3.3.1. White Team Notification

The White Team is a hierarchy of simulation engineers, analysts and test conductors whose mission it is to insure the proper and valid execution of the simulation hosting the virtual experiment. The White Team, unlike the Blue Team, Red Team, Green Team and Gray Team, is the only team to have access to truth data in the virtual experiment. The other mentioned teams only have access to perceived data which they obtain through their various sensors in the simulated world. The Blue Team is comprised of typically friendly forces and contains the system or systems under test for the experiment. The Red Team is the opposition or enemy forces. The Green Team, sometimes referred to as culture, is independent, non-biased inhabitants of the world. The Gray Team is entities whose affiliation or allegiance is unknown.

Synthetic entities as well as manned entities can often behave in unpredictable ways during a virtual experiment. Unpredictability is the essence of virtual experimentation and is often a good thing except when it crosses over the defined boundaries of the experiment and threatens to invalidate the results. There are many different types of this behavior that could be detected early enough by White Team Notification (WTN) agents and corrected within the boundaries of the experiment and subsequently keep the results of the experiment valid.

Unrealistic behavior by synthetic entities is a concern. Synthetic entities not reacting correctly or even at all to an enemy threat is a typical example. Although these behaviors may sometimes be hard to detect it is the unexpected behavior by manned entities that can often prove to be the most challenging to correct. How is unexpected behavior by a manned entity not desired? The operators are human after all. The unexpected behavior is not desired when it crosses outside the boundaries of the experiment domain. One example of this kind of behavior is traversing outside the geographic boundaries of the simulated world and “falling off the map”. Nothing breaks the suspension of disbelief of a human participant more than finding themselves in an area or region with no terrain or geographic markers. Although “falling off the map” refers to being geographically placed outside the simulated world, crossing outside the boundaries of the experiment domain is not just a spatial or geographic term. Undesirable behavior can just as easily lead to invalid missions, roles, affiliations, communications, etc.

“Falling off the earth” and other instances of crossing the boundaries of the experiment domain are usually correctable if caught in time. The key is to be able to predict when it will happen well before it does happen. An example during a virtual experiment of a Suppression of Enemy Air Defenses (SEAD) mission is given. A strike aircraft is given the mission to hit selected targets but is also tasked to hit perceived high priority pop-up targets. The simulated world in the experiment covers a large portion of the enemy territory but not the entire territory. The strike aircraft due to miscommunication could realistically believe there is a high priority pop-up target in the section of enemy territory that is not a part of the simulated world for the experiment. Due to the miscommunication, the strike aircraft’s forward air commander is also not aware of the mistake and thus will not call him off. If the strike aircraft is allowed to cross over into the territory that is not in the simulated world then the experiment will be deemed invalid and time and money will be wasted.

WTN agents can be inserted to reduce the number of human operators on the White Team that are needed to detect and resolve these undesired occurrences. WTN agents use subject matter expert strategies to search the simulated space and detect undesirable and desirable behavior by synthetic or manned entities. Although not referenced in this SEAD strike aircraft example, desirable events are also of interest to the White Team for any number of reasons that influence the consistent execution of the simulation.

WTN agents have access to truth data and perception data as well as communications to and from all entities. Due to the immense size of the environment, a series of thresholds need to be broken before the White Team is even made aware of a possible desirable or undesirable event. When notifying the White Team, the WTN agent supplies them with as much information as possible about the event as well as possible resolutions if available. In more mature systems, the WTN agents may autonomously resolve the event if the probability of resolution is scored high enough.

In the SEAD strike aircraft example a number of resolutions may be available if detected in time. By comparing the trajectory of the strike aircraft with the known simulated space and detecting certain keywords and phrases, the WTN agent determined that the strike aircraft’s future trajectory would take it out of the simulated world of the virtual experiment. One example of a solution would be for a White Team member using a WTN agent-driven user interface to detect and classify the event and then notify the White Team leader to insert a communication into the command and control chain to re-task the strike fighter. The inserted communication would need to be inserted at an appropriate level of the command and control chain to keep any human participants in the test from noticing anything wrong or strange with the scenario.

An alternate resolution to the SEAD strike aircraft scenario is to quickly “grow” a simulated region outside the original simulated space so the strike aircraft would not notice anything wrong when veering into that area. In more mature systems this resolution may be applied
autonomously by a WTN agent. Once the WTN agent detected the undesirable event it would automatically task an ASG agent to grow the area of the world based on region data. This ASG agent may be like the ones described in the Automatic Scenario Generation section of this paper. Of course, in order to generate or “grow” the new simulated space, computer resources would be needed requiring a CLO agent to be tasked to reallocate the CPU loading and give consent to the resolution. All tasking and manipulation of the simulated world would most likely be applied through human management by exception. The goal of the system would be to provide high probability solutions to undesirable events to insure proper execution of the simulation during congested and high activity periods in which White Team overload is a possibility.

3.3 Knowledge Extraction

A virtual experiment is only as successful as the information and data gathered during the execution of the simulation. Although some information is learned by the human participants in the experiment, the predominant portion of the analysis and results are obtained through knowledge extraction on the logged data during and after an experiment run.

A typical week long experiment may consist of 40 test runs, 50 manned entities, 2000 synthetic entities and a White Team of 30 engineers, analysts and conductors. Trade-offs are constantly made between execution time and turnaround time to efficiently complete the test while still meeting the test objectives. Obtaining and applying knowledge from previous runs to outstanding runs is a key enabler to efficiently and thoroughly filling out the test matrix of the experiment. Due to the immense amount of data gathered and the immediate need for some of the results, knowledge extraction is another excellent area for the implementation of intelligent agents in virtual experimentation.

Knowledge extraction is the domain of the operations analyst. Many of the strategies used by analysts can be represented in knowledge extraction (KE) agents. These KE agents can process large amounts of real-time or logged data. KE agents are designed to traverse and navigate worlds of data including multiple dimensions of relational and networked databases. Benefit is obtained from being able to visualize the complex and hidden relationships and interactions between entities and their behavior in a virtual experiment hosting a complex adaptive system. KE agents will aid in the cumbersome job of preparing, processing and visualizing data to help illuminate these relationships, concepts and ultimately knowledge.

KE agents would be tasked to automatically calculate the values of key measures of effectiveness (MOEs) and measures of performance (MOPs) during the real-time execution of the experiment. The MOEs and MOPs are often immediately used to alter the remaining test runs of the experiment to make better use of remaining time and budget. Many values of first order MOEs and MOPs are easily obtained using standard data mining techniques. KE agents give the ability to extract second and even third order cause and effect relationships and their metrics. KE agents are especially efficient at gathering contextual metrics and detecting or identifying sequences of events that are of interest for future experimentation.

Does the information the analyst collected contain the knowledge he was after when preparing the experiment? Or is that knowledge derived from further processing of all available information. The values or “numbers” are not usually the ultimate goal of experimentation but rather it is the semantic or logical knowledge that produces those metrics. KE agents can help illuminate and identify this knowledge. Can KE agents be used to tell the analysts what they should be looking for but aren’t? Can they give suggestions on how to better design the experiment to more effectively obtain the knowledge the analysts desire? The answers to the above questions and many more will define the intelligent knowledge extraction tools of the future.

3.3.1 Visualization Prioritization

KE agents are also used for Visualization Prioritization. Visualization Prioritization is needed to detect and prioritize events for a demonstration to a VIP audience. One effect of the unpredictability of virtual experimentation is not having a scripted engagement or sequence of events that would be of interest to a VIP always available. After all, no one knows when anything is going to happen for sure during a large virtual experiment. KE agents would be instantiated to detect events of interest based on the rules and strategies of analysts that are knowledgeable of the demonstration needs of the experiment. These KE agents would run during real-time execution, detect many possible occurrences of the event of interest and then intelligently process and visualize this information through a KE agent-driven user interface. In a simulation with 10,000 entities the number of missile engagements or radar detections can be quite numerous at any given point in time. There may be a very high probability of a certain type of missile engagement or radar detection during a period of time but knowing for sure when and where it will occur and that the outcome will be favorable to the particular VIP of interest is another matter.

4. CONCLUSION

Extreme value in virtual experimentation is obtained from emergent and synergistic benefits of large-scale environments. Only recently have advances in technology enabled the modeling of a high enough number of entities and synergistic systems at a certain level of fidelity to produce emergent behaviors and interactions. The benefits of these complex adaptive systems have begun to outweigh the cost of their production. The capability to insert virtual experimentation into all levels of a products life-cycle
provides huge benefits in the areas of engineering and cost. The emergent benefits of a complex adaptive system enable the customer to not only use virtual experimentation for product validation but also to discover unknown requirements and design concepts during the early engineering phases.

Producing the emergent and synergistic benefits in virtual simulation is highly dependent upon a tight relationship between subject matter experts and the modeling of the intelligent agents. This tight relationship drives the need for behavior matching instead of textbook theory matching. New software development tools will be necessary to generate intelligent agents that behave and learn like humans based on rules and strategies extracted from subject matter experts.

Other key areas for future focus are the role of learning and adaptation in intelligent agents and the pros and cons to increasing their level of autonomy. Humans are constantly learning and adapting at a high rate. Humans are also completely autonomous entities. Will learning and adaptation give us more or less confidence that intelligent agents will behave as desired? Will learning and adaptation be constrained to the intelligent agent design phase and not be applicable during real-time execution of a virtual experiment? These questions and other like them will be answered as virtual experimentation and large-scale simulation environments grow in complexity and scope.