Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

USE OF INTELLIGENT CONTROLLER NODES TO AUGMENT HUMAN ROLE-PLAYERS IN SYNTHETIC BATTLESPACE EXERCISES

Gary E. Buker

Raytheon Company
50 Apple Hill Drive
Tewksbury, MA 01876, U.S.A.

Brian C. Flis

Air Force Research Laboratory
2698 G Street
Wright Patterson AFB, OH 45433, U.S.A.

ABSTRACT

Distributed exercises, which combine live, virtual, and constructive entities, are increasingly used for training, readiness exercises, and CONOPS Analysis. These exercises require the support of human controllers who monitor and control simulated entities to maintain realism of the exercise. The high cost in time, money, and personnel for supporting large exercises makes it important to reduce the number of role players required. In addition, the large volume of information flowing from the synthetic battlespace and the narrow window for response dictate a need for mechanisms to assist the human role players who control the simulated entities. The Role Player Intelligent Controller Node (RPICN) uses intelligent agents to monitor and interact with the simulation environment. The agents can recognize significant events, make recommendations, and perform routine actions for the role player. By increasing situational awareness and automating routine tasks, the RPICN improves the efficiency of the human controllers.

1 INTRODUCTION

The use of simulation technology can be found in most areas of the military. Simulations are used to train personnel at all levels. Often simulations are run jointly to train personnel in a broader spectrum of single and inter-service doctrine, tactics, techniques and procedures. The Air Force, for instance, uses the Air Warfare Simulation (AWSIM) to train commanders in the Air Operations Center (AOC) during joint simulations. In most cases, these simulations require a large number of support personnel, or role players, to control the simulations in order to meet the training goals. The realism of the training is highly dependent on the background, skill level, and abilities of the role players.

Technical controllers/role players are those individuals who set up and run an exercise. During an exercise, technical controllers/role players control the exercise via exercise parameters. In this paper, technical controllers are the computer professionals who control the mechanics of starting, stopping, and keeping the exercise running. Role players are domain professionals who participate in the exercise in the context of a specific role that is organizationally outside the roles being trained. Blue Force role players often use real-world C3 systems to communicate with the training audience and to generate game inputs. For instance, role players may receive Air Tasking Orders (ATOs) from the AOC. In the case of AWSIM, utilities exist to automatically translate the ATOs into AWSIM pending orders. Although role players are responsible for reviewing the automatically generated missions, most of their responsibility is to monitor the simulation game in order to assure missions are carried out to meet the training goals. Opposing Force (OPFOR) role players often do not have the benefit of automatically generated mission plans. They often perform the time consuming task of generating the opposing force mission plans by hand and entering them into the simulation manually. The background and skill level of the role player highly effects the level of realism of the opposing force. Thus, the success of the exercise depends on the availability of trained role players on both sides.

Due to the high cost in time, money, and personnel of running training exercises, the Air Force has a goal of reducing the number of human role players required in these endeavors. In addition to reducing personnel, the large volume of information flowing from the synthetic battlespace and the narrow window for response dictate a need for a more efficient processing mechanism to assist the human role players who currently run the exercise. This paper will discuss the approach used by the Role Player Intelligent Controller Node to address these issues.

2 OVERVIEW

In the Agent-based Modeling and Behavioral Representation (AMBR) program, researchers in the Air Force Research laboratory (AFRL) Human Effectiveness Directorate are developing new approaches to simulate intelligence in order to meet Air Force needs. In particular, they are developing and demonstrating agent-based approaches to emulate intelligence. Intelligent agents, a software process designed to work asynchronously and autonomously in a particular environment, can "perceive and interact" with the world. Further, they possess agency, the ability to proactively engage in goal-directed behavior. Agent developers normally provide computational primitives that enable an agent to monitor a data stream, looking for patterns of information. The perception of specific patterns activates goal-directed activity. More advanced agents may embody sophisticated reasoning strategies. These agents can evaluate alternative plans of action and choose the best one to execute. Researchers are developing highly sophisticated agents, which can dynamically alter plan execution in real-time based upon changes in the environment.

The AMBR program consists of a series of demonstrations occurring along two tracks: practical demonstrations (using current technology) and emergent development (creating new modeling techniques). In the practical demonstration track, a relatively mature technology is applied to a specific problem. The goal in this track is to solve specific problems of interest to the user and acquisition community in time for the solution to be incorporated into a major acquisition program.

As part of this track, Raytheon created a Role Player Intelligent Controller Node (RPICN), which is an autonomous agent-based model capable of "seeing" changes to the battlefield and reporting them back to the role player. The RPICN can also provide the role player with enhanced capabilities to control the new agent-based models. The models are more intelligent because they are able to see and react in real-time to changes on the battlefield. This agent-based capability enables designers to program more sophisticated behavior (i.e., if you see a SAM, go around it). In addition, a role player's workstation provides the human role player enhanced situational awareness of all the models the role players are controlling and tools to modify agent behavior in real-time (i.e., as it is flying the mission). The ultimate goal is to enable a role player to control more aircraft, in a more efficient manner, while providing more realistic behavior for the simulated forces under his control.

3 CAPABILITIES

The RPICN interacts with the synthetic battlespace to gain situational awareness and interacts with the role player to support role player training objectives. The RPICN can assess the synthetic battlespace based on goals established by the role player. Based on these goals, the RPICN assists the role player with maintaining game realism. Figure 1 depicts the relationship between the RPICN and the role players' working environment.

3.1 User Interface

The role player interacts with the RPICN through a graphical user interface (GUI) on his normal workstation. Each role player has a unique graphical user interface displaying only data and information associated with his logon

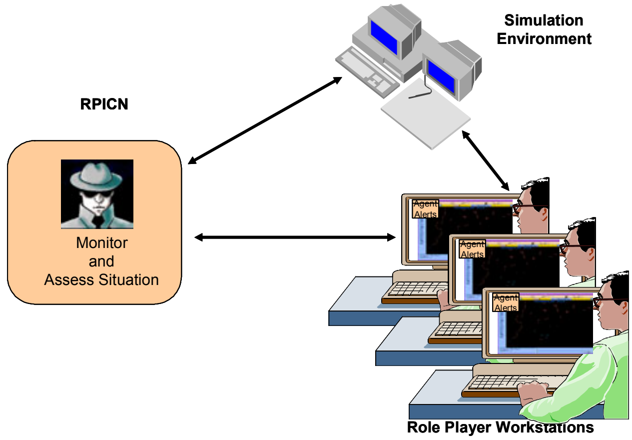

Figure 1: RPICN Environment

session. In the AWSIM environment, many of the RPICN role player capabilities can also be accessed directly from the Graphical Input Aggregate Control (GIAC) interface, as well as from the RPICN windows. This feature allows the RPICN to integrate with minimal interference into the role player's normal activities.

Through the GUI, the role player has the ability to initiate and monitor the status of software agents, define message filtering and prioritization parameters, as well as specify the level of assistance desired. The RPICN provides the role player with the ability to save and restore RPICN session parameters established by the role player. This allows the role player to reuse established preferences and parameter data.

3.2 Objectives

Within the RPICN, the role player assistance capabilities are referred to as "Objectives". Each Objective defines a behavior tailored to provide specific situational awareness and assistance to the role player. The Objectives notify the role player of a situation, and in certain instances will recommend a Course of Action (COA). There are three Levels of Assistance (LOA) that the role player can choose from, as shown in Table 1.

Table 1: Levels of Assistance

LOA	Actions
1	Notification only. The situation is identified for the user, but no COA is recommended.
2	The situation is identified for the user, and a COA is recommended. The user can choose whether to act on the COA.
3	The situation is identified for the user, and a COA is displayed. The COA is automatically executed as soon as it is identified.

Each instance of an Objective can also be assigned a priority level by the role player, as shown in Table 2.

Table 2: Priority Levels

Priority	Actions
High	The user is alerted of a situation by a pop-up window containing the text of the alert, a course of action (if applicable), a dismissal button, and a confirm button (if applicable).
Low	The user is alerted of a situation by the word "alert" in the RPICN Role Player Client display. If desired, the user can view the full text of the alert and its COA.

The following sections name the Objectives that were developed for use with AWSIM, describe the situations they were intended to address, and discuss the assistance and recommendations they provide to the role player.

3.2.1 Monitor Air Terminal Events

The AWSIM Air Terminal (AT) is a textual display that prints out a record of significant events as they occur. In a large exercise, the number of events scrolling by on the AT can be quite large. Since most role players are interested in only relatively few flights or events, monitoring the AT can be a tedious process that results in missed events.

The Monitor Air Terminal Events Objective assists in this situation by monitoring for events that are significant to the role player. The role player can designate what events are of interest for specific flights. When one of these events is detected, a pop-up window appears alerting the role player of the event. These notifications are advisory only, with no recommended course of action. Figure 2 depicts the functionality of Monitor Air Terminal Events.

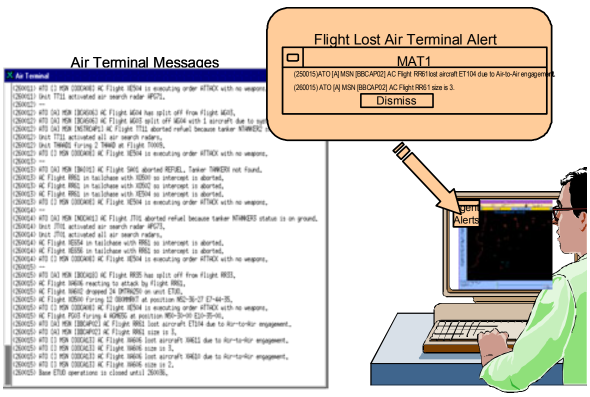

Figure 2: Monitor Air Terminal Events

3.2.2 Monitor Target Coordinates

In distributed exercises involving multiple simulations, incorrect attack coordinates can sometimes be entered. This can happen for several reasons. The coordinates may not correlate between different simulations, the target may have moved since the coordinates were obtained, or the operator could have made a typographical error. If the coordinates for a target in an attack command do not correlate with a target of the designated type, the target will not be hit. AWSIM can correct for an error of up to three miles, but if the target is off by any more than that, the incorrect coordinates are attacked.

The Monitor Target Coordinates (MTC) Objective searches the flight's pending orders to correct for such inconsistencies. The pending orders constitute a flight plan, which is broken up into "legs." After receiving these orders, the objective analyzes them for an attack leg. If an attack leg is found, the objective searches the AWSIM environment for a target within three miles of the coordinates given in the orders. If the target's coordinates match the ones in the orders, the next attack leg is looked for. If, however, the coordinates do not contain a target of the given type (e.g. radar, SAM, airbase), then the objective searches the simulation for an alternate target of that type within a user specified search radius and recommends it for targeting. If the role player accepts any of the recommended changes, a reflight-plan command is issued to AWSIM to update the pending orders with the recommended coordinates. Thus this objective serves as both a checker to validate the orders that were entered, and an advisor to make corrections.

3.2.3 Monitor Fuel

All attack missions are given a set of orders before taking off from their base. They also should be initialized with enough fuel to complete whatever mission they are assigned. Occasionally, however, a flight may have to deviate from its original course or change its orders to compensate for an unanticipated need. When this happens, it can cause the flight to have insufficient fuel to complete its original mission. If AWSIM detects that a flight is out of fuel, it will order a "bingo," which will return the flight to the nearest friendly base.

The Monitor Fuel Objective assesses the fuel requirements for a Flight of interest, keeping in mind the current levels of fuel in the aircraft. It performs this action periodically throughout the mission, similar to a pilot in the real world. If it detects that there will not be enough fuel to complete the scheduled mission, the Objective recommends replenishing the needed amount, thus preventing the Flight from running out of fuel and being transported to the nearest friendly airbase. Although there may be logistical issues with "magically" replenishing fuel, there may be situations where the role player may have the authority to assure the success of the mission.

3.2.4 Sam Site Avoidance

When an entity receives attack orders, it does not check to see if its flight path will place it within range of a Surface-to-Air Missile (SAM) site. This can have unexpected consequences, particularly if the training goals of the exercise depend on the entity accomplishing its mission.

The Sam Site Avoidance (SSA) Objective assesses the pending orders for the Flight of interest and identifies SAM sites that it will pass. If the Flight is threatened by a SAM site, the SSA Objective tries to calculate a course around the range of the SAM site. If there is a way around, it then checks to see if the new route lies within a different SAM site's range. If so, it tries a different approach. SSA calculates the Line of Sight (LOS) of the SAM site, decrementing the aircraft height by 100' each time. It continues to do this until the LOS is less then the range of the SAM site. The role player is notified of the situation and the recommended COA. If the recommendation is accepted, a manual command will be entered by the RPICN to route the flight around the SAM site, after which the flight will resume its planned mission, as depicted in Figure 3.

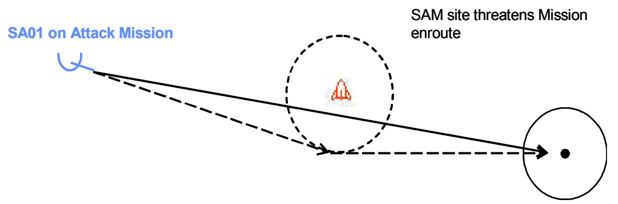

Figure 3: Sam Site Avoidance

3.2.5 Restrict Pairings

The Restrict Pairings Objective assesses all threatening Flights located within a role player specified area of interest. Based on this assessment, the objective recommends that a Flight pair against an enemy based on proximity and heading. The area of interest can be a radius around the entity, or a rectangular area of interest. If an enemy violates the area of interest, a pairing is recommended.

3.2.6 Cascading Efects

Cascading Effects (CE) are secondary effects that are not directly tied to a first order physical event. An example of a first order event (a.k.a. first order effector) could be a combat Flight entering restricted air space. The entering of the air space would be the first order effect. There could also be secondary effects, not currently modeled. For example, if the air space is a neutral country's, its violation could set off a sequence of political and social reactions resulting in the country deciding to deny access to supply bases or permission to overfly other airspaces. These second order effects would likely be exercise-specific (i.e., related to specific exercise goals), and not part of the Synthetic Battlespace (SB) simulation models.

Secondary effects are the result of interactions within and among political and social entities that must be defined for the specific simulation exercise. The user defines the characteristics of these entities and the rules that describe their behaviors and first order effectors prior to initiating the RPICN. These user inputs collectively constitute an AMBR Cascading Effects Knowledge Base (ACE KB). The RPICN maintains an internal representation of the SB entities based on the parameters provided by the role player. These parameters include both options selected by the role player at runtime and the user-defined knowledge base that is specified before the exercise begins. When the RPICN identifies a first order effector in the SB simulation, it evaluates the potential sequence of Cascading Effects by applying the ACE KB to the current situation. The RPICN provides the role player with a list of secondary effects (consequences), if any, from which to select a preferred course of action (COA).

If the role player selects a consequence, the RPICN updates its internal representation based on the first order effector. It continues to monitor the SB simulation and provides the role player with on-going status as the cascading events associated with the selected consequence develop over time. When the internal representation of cascading events produces a secondary effect, the RPICN takes the appropriate action to reflect that secondary effect back to the SB simulation (e.g., insert an interaction directly into the simulation, notify the role player to take an action). For the initial implementation, the RPICN requires role player approval for all secondary effects, i.e., Cascading Effects use only Level 2 LOA.

4 ARCHITECTURE

The RPICN is composed of three components that normally reside on three separate machines – the Agent-based Modeling and Behavioral Representation Artificial Intelligence Node (AMBRAIN), the Role Player (RP) Client(s), and the Joint Synthetic Battlespace (JSB) Client. The AMBRAIN contains the agents that model the behaviors of the role players. It operates within the OMAR behavior-modeling environment. The RP Client(s), which runs on the role players' workstation, provides the user interface through which role players interact with the AMBRAIN. The JSB Client provides the interface to the JSB through which the AMBRAIN obtains simulation data and interacts with the JSB on behalf of the role player. This configuration is shown in Figure 4.

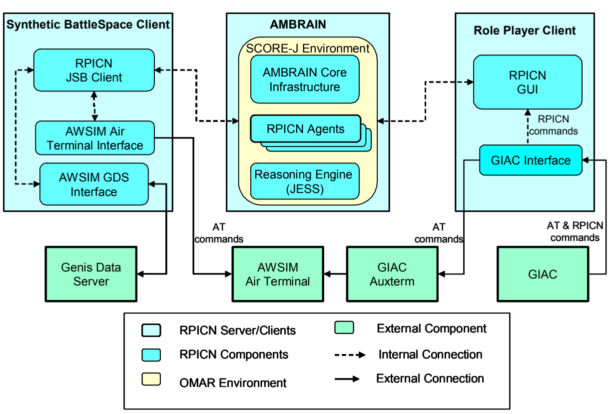

Figure 4: RPICN Architecture

4.1 AMBRAIN

The AMBRAIN is the component of the RPICN that performs the role player behavioral modeling. It is implemented using the OmarJ environment, which provides the infrastructure for building agents, defining their goals, and controlling their interactions. OmarJ is an agent development environment that provides tools for creating and managing systems of agents. OmarJ provides a procedural language for defining behavior (ScoreJ), a time management component, an inter-process communication mechanism, an event recording mechanism, and an external communication layer. It provides an object-oriented framework for defining simulation objects as Java classes

The AMBRAIN also incorporates the Java Expert System Shell (Jess), which is a tool for building a type of intelligent software called Expert Systems. An Expert System is a set of *rules* that can be repeatedly applied to a collection of *facts* about the world. Rules that apply are *fired,* or executed. Jess uses a special algorithm called *Rete* to match the rules to the facts.

The agents in the AMBRAIN fall into two classes – Manager agents and Objective agents. Manager agents are persistent agents that manage the resources used by the AMBRAIN. For instance, there is a Data Manager agent that maintains the truth data received from the simulation environment and distributes it to other agents that need it. The Dispatcher agent allocates and de-allocates Objective agents to implement the goals of the role players. Objective agents are agents that are dynamically instantiated to implement a specific instance of an Objective that a role player has requested.

4.2 RP Client

The Role Player Client provides the graphical user interface through which the role player interacts with the RPICN. The role player has the ability to logon and logoff the system without effecting other users of the system. Each role player has a unique graphical user interface displaying only data and information associated with their logon session. The RP Client interfaces to the GIAC through the GIAC auxterm. Output from the GIAC to the auxterm is intercepted and filtered by the RP Client. If the output contains a special header associated with the RPICN, it is routed to the RP Client for processing. Otherwise it is forwarded to the auxterm as usual. In this way, command buttons can be defined on the GIAC to directly initiate commands to the RPICN.

4.3 JSB Client

The Joint Synthetic Battlespace (JSB) Client provides the RPICN with an interface to the simulation, specifically to the AWSIM Air Terminal (AT) and the Genis Data Server (GDS) used by AWSIM, as well as the JSIMS HLA Federation. The purpose of the JSB Client is to isolate simulation specific interface implementations from the agent implementations. This makes it easier to adapt to interface changes or to include additional simulation interfaces.

5 CONCLUSIONS

Intelligent agent systems such as the AMBR RPICN are useful tools for assisting role players to identify and correct situations that may lead to an undesirable result. They also allow the role player to focus on the larger issues of making decisions on the progress and direction of the war game, while freeing him from tedious details, information filtering, and the possibility of missing important and often unanticipated developments

The users of the simulation benefit in multiple ways from AMBR RPICN support. They are freed from having to spend inordinate amounts of time gathering information, and can use that time to make important training decisions. They are also relieved of the worry of missing an important occurrence due to a plethora of potential interruptions. Whatever the role players are doing, the AMBR RPICN can always be monitoring for pertinent developments in the game environment. The role players themselves have expressed an interest in having entities that can monitor a game at night or at other times when an operator would be away from the station. Another benefit is to the trainee. If mistakes and omissions can be reduced, then the trainee will, ostensibly, receive a better and more accurate training session. AMBR RPICN gives the role players better control over the simulation environment, which in turn allows them to focus on the proper training situations.

AUTHOR BIOGRAPHIES

GARY BUKER is a Principal Software Engineer for Raytheon Company in Tewksbury, Massachusetts. He has a B.A. in Mathematics and Physics from Dartmouth College. His research interest are distributed simulations, agent technology, and artificial intelligence. He is a member of IEEE. His e-mail address is `<Gary_E_Buker@raytheon.com>`.

BRIAN FLIS is a Simulation Research Officer for the Human Effectiveness Directorate of the Air Force Research Laboratory. He has a B.S. in Mechanical Engineering from the United States Air Force Academy and a M.S. in Mechanical Engineering from the Georgia Institute of Technology. His e-mail address is `<brian.flis@wpafb.af.mil>`.