IST Virtual Environment Team Training System, Intelligent Tutoring Enhancement: Final Report

1-1-2001

Glenn A. Martin

INSTITUTE FOR SIMULATION & TRAINING

January 16, 2001

FINAL REPORT

University of Central Florida

IST-CR-01-15

IST Virtual Environment Team Training System – Intelligent Tutoring Enhancement

Final Report

IST-CR-01-15

II

January 16, 2001

Institute for Simulation and Training
University of Central Florida
3280 Progress Drive
Orlando, Florida 32826

INSTITUTE for SIMULATION & TRAINING

Submitted To:

Research Development Corporation
2875 Towerview Road, Suite A4
Herndon, VA 20171

IST Virtual Environment Team Training System – Intelligent Tutoring Enhancement

Final Report

Submitted By:

Glenn A. Martin
Senior Research Scientist
407-882-1349
martin@ist.ucf.edu

INSTITUTE for SIMULATION & TRAINING

Institute for Simulation and Training
University of Central Florida
3280 Progress Drive
Orlando, Florida 32826-0544

IST Virtual Environment Team Training System – Intelligent Tutoring Enhancement

Final Report

Written By:

Glenn A. Martin

January 16, 2001

Abstract

The principal purpose of the IST Army Research Institute Virtual Environment Research Testbed is to provide a flexible and adaptive mechanism which can be used to conduct behavioral research on factors which affect the acquisition of skills in a virtual environment and the transfer of those skills to the real world.

Research conducted at IST under contract with the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) has been successful in identifying virtual environment interface requirements for training applications, documenting the effectiveness of virtual environments as a training medium, and researching the transfer of skills acquired in a virtual environment to the real world.

This report describes work performed between March 2000 and December 2000 for Research Development Corporation to integrate their intelligent tutoring system to the ARI Virtual Environment Research Testbed.

1. Introduction

The U. S. Department of Defense has made a major commitment to the use of networked real-time simulators for combat training. A principal component missing from today's training simulations on the electronic battlefield, however, is the dismounted soldier. Little is known about the effective use of these virtual environments for training infantrymen. To address this issue, the U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) and the University of Central Florida Institute for Simulation and Training (IST) have established the ARI Virtual Environment Research Testbed at IST's Visual Systems Laboratory. The goal of the Testbed is to investigate the use of virtual environment technology for Army training and develop training strategies and performance measures to make virtual simulations an effective training medium for dismounted soldiers.

The ARI Virtual Environment Research Testbed, established in the summer of 1992, has been successfully designed and developed for purposes of research and experimentation in the application of virtual environments to military tasks for the dismounted soldier.

The Testbed provides a flexible and adaptive mechanism, which can be used to conduct behavioral research on factors that affect the acquisition of skills in a virtual environment and the transfer of those skills to the real world. Working together, ARI research psychologists plan, design and conduct the experiments and analyze the data, and IST researchers, mostly software and hardware engineers, develop the necessary models, hardware, and software simulation components.

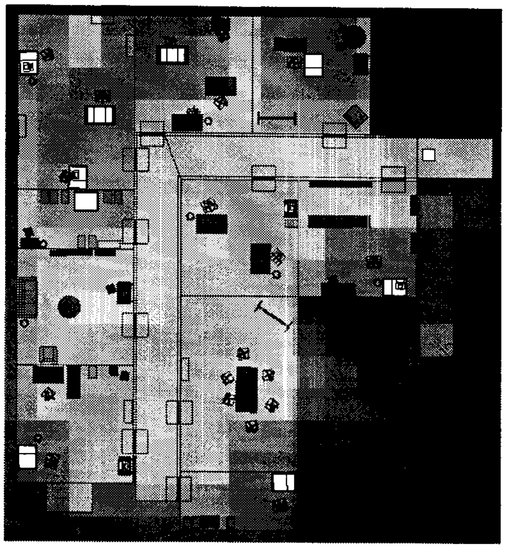
This report will first provide an overview of the testbed, followed by a description of the work performed to merge in Research Development Corporation's (RDC) intelligent tutoring system.

2. The Virtual Environment Team Training System

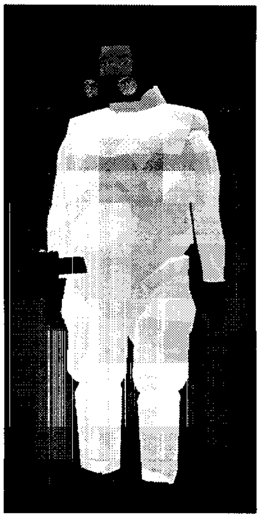
The IST Virtual Environment (VE) Team Training System, developed at IST over the past several years, has been brought to a sophisticated level. This system can support local and distributed team players. It can support a broad range of input devices, output devices, tools, trackers, etc. As an example of what this system of software can currently support, one team-training scenario is described below. The hardware and software used to support this scenario will be enhanced by the proposed tasks described in a future section.

The training scenario is a building search exercise with a two-person team under resource, time, and environmental constraints. The trainees are a team leader and an equipment specialist. The building is searched for targets (chemical canisters) that have to be neutralized within a certain time. The building has normal complex rooms and areas that serve to conceal the targets. The scenario also has opposing forces that can act to distract the trainees and increase the complexity of the scenario.

A typical 10-room building floorplan is pictured to the right.



Typical 10-Room Building



Team Member Avatar

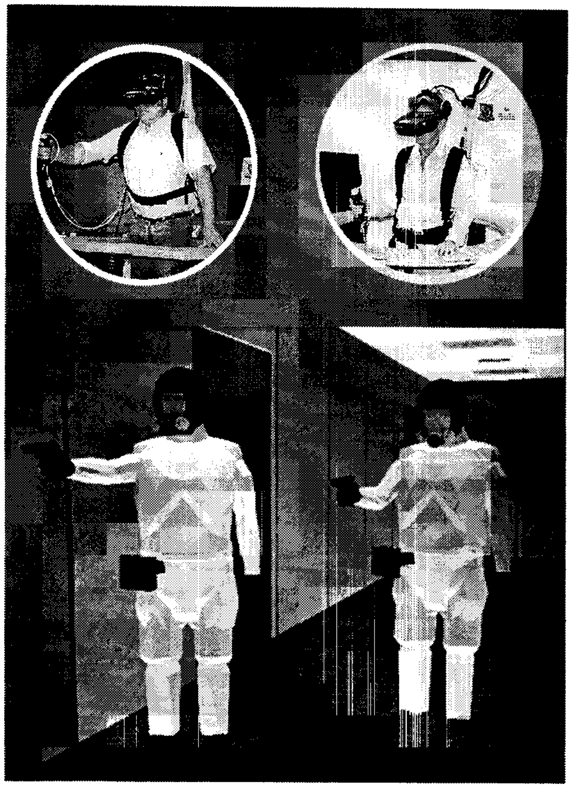
Each of the two simulated team members wear protective clothing, a clock that shows the remaining airtime in the simulation, and a breathing apparatus to prevent exposure to the toxins. See picture of avatar wearing chemical suit with gear to the left.

The physical confinements of wearing immersive VR gear (helmets, tethers, etc.) actually enhances the immersion value of the situation since the protective gear similarly limits vision, locomotion, and audition.

The team must deal with the computer-generated enemies, innocent bystanders, and leaking gas canisters as they move through the building.

The team leader and equipment specialist must cooperate and communicate to efficiently search for gas canisters, detect and identify opponents as quickly as possible, and search the maximum area within the timeframe of the exercise.

The bottom portion of the picture to the right shows the two-player team in the virtual environment. The top portion shows the two corresponding live players.



Two Player Team

The simulation requires a system of sensors for each team member, a head mounted display with stereo view, and a joystick device placed in the right hand that allowed the players to change their tools and trigger them.

The system also uses a simulated radio net that allows the team members to communicate with each other and the mission commander.

Audio cues like footsteps, gun shots, and doors opening and closing are used in the simulation to help the team members feel more immersed.

Six tracking sensors are used to track each trainee's position and orientation in the environment.

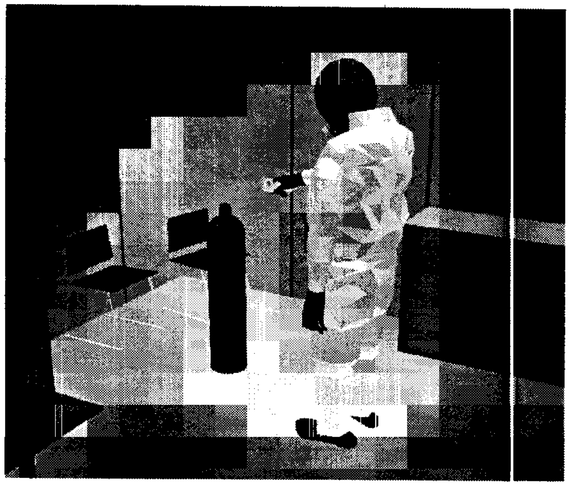
- A sensor mounted to a backpack controls body orientation.
- A sensor on each leg allows forward movement through a natural marching movement and backward movement by taking a step back.
- A sensor mounted to the helmet controls the view displayed inside the head-mounted display so a player can look around.
- A sensor placed at the elbow of the right arm and on the hand-held device allows a player to gesture, aim and fire weapons, and operates equipment in the virtual environment.

Using these sensors, the simulated system is able to fully articulate each player in the virtual environment.

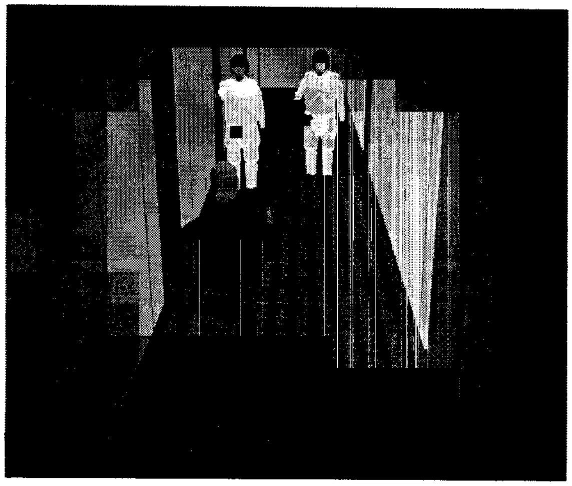
All mission activity, including sounds and communications, are recorded during the live mission runs. The entire mission can be viewed flat-screen using a VCR-like control to allow real-time playback as well as frame-by-frame, fast-forward, rewind, or slow motion.

The system supports running local teams as well as remotely distributed teams. Communication for remote teams is handled over ISDN.

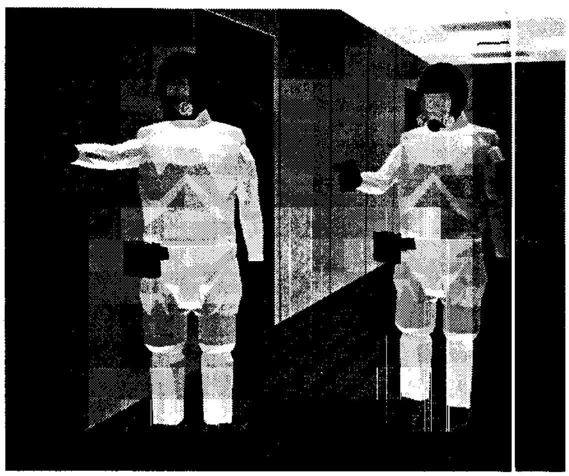
Stealth views are supported to allow others to view the scenario from a flat-screen. During the actual scenario described above, one additional player, the Commander, uses a Stealth to oversee the mission as it is taking place.



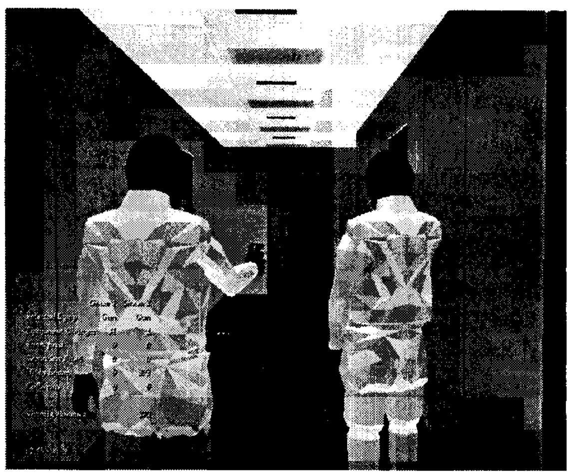
Equipment Specialist using Canister Detection Device



Team Approaching Enemy



Team Stacking Procedure in Hallway



Stealth/Commander View

3. Integration of the RDC Tutoring System

There were two possible methods for integrating the RDC tutoring system into the ARI/IST FITT system. The first was to have the FITT system directly interface over a network with a PC program using the RDC libraries. The second was to write a general-purpose application on the PC that would work with any DIS simulation. The first solution would have been a specific solution while the second is a more general solution. We chose the latter solution as it best supports current needs as well as leaves open the possibilities of future needs.

The PC Interface Application

The PC Interface Application reads DIS packets from the network and gives the appropriate information to the RDC libraries. This is a simple summary, but it actually consists of three important steps: initialization of the RDC libraries, getting data to the PC application, and getting data to/from the RDC libraries.

Initialization of RDC Libraries

In order to iniitalize the RDC libraries, a fair amount of work is necessary. To begin, a series of classes that inherit from abstract RDC classes must be built. For this work, we built the following classes (a short description of each is given):

`ISTDoor`	Contains information about a single door within the environment
`ISTFireTeam`	Contains information about each fireteam in the scenario
`ISTRoom`	Contains information about each room within the environment
`ISTScenarioInfo`	Contains information about the scenario
`ISTSoldier`	Contains information about a single soldier within the environment
`ISTTeammate`	Contains information about a soldier's teammate

In addition a class to represent the outline of each room was created. This class is known as `FootprintPolygon` and basically provides a test to determine if a soldier is within a room.

In order to make the instantiation of each class necessary, support for a configuration file was included. This configuration file provides complete data about the scenario including all doors, rooms, room footprints, fireteams and soldiers. Below is a sample configuration file:

```
# door <room_id> <door_id> <num_locks> <num_hinges>
door 1 1 1 2

# room <room_id> <door_id>
room 1 1

# footprint <room_id> <x,y vertices...>
footprint 1   0 0 10 0 10 10 0 10

# fireteam <team_id>
fireteam 1

# soldier <dis_site> <dis_host> <dis_entity> <fireteam_id>
soldier 1 1 1 1
```

Once all of the classes are instantiated based off the information in the configuration file, the RDC Reasoning Engine can then be initialized with a call to `CSTRICOM_Reasoner::RDC_Init()`.

Input to System

With the RDC libraries initialized, data can then be fed to the reasoning engine. In this case, the data is coming from DIS PDUs. In this work, three PDUs were supported: the Entity State PDU, the Detonation PDU and the Action Request PDU.

The Entity State PDUs are the most complicated to process. They are used to feed the internal dead reckoning of the IST application and must be properly converted out of geocentric coordinates. Since DIS dead reckons entities, for the RDC libraries to know where each entity is located, they must be dead reckoned. Therefore, the IST application performs the dead reckoning and passes the estimated positions for each entity to the RDC libraries using an `OV_MOVEOUT` Occurrence Trigger. Note that the DIS Entity ID from the PDU must be matched against the instantiated `ISTSoldier` objects. Furthermore, the "current room" in the `CSTRICOM_Sginfo` structure must also be updated.

For the Detonation PDUs, the IST application must determine if the detonation was caused by a bullet being fired from a gun or by a splat being fired from the paint marker. If the actual gun is being used, the IST application created an `OV_KILL` Occurrence Trigger and tells the RDC libraries. In the IST FITT application, if any entity is hit, it is killed; there is no concept of injuries currently. If the paint marker was the tool being used, the IST application creates an `OV_MARK` Occurrence Trigger and notifies the RDC libraries appropriately.

Lastly, if an Action Request PDU is received, the IST application creates an `OV_BREACH` Occurrence Trigger. While door openings are not technically provided for in the DIS Standard, IST has been using the Action Request PDU for this purpose.

It is important to note here that as other DIS PDUs are desired to be supported, they can easily be added to the IST application. Indeed, the framework is already in place by use of empty "process" methods in the IST Application.

<u>Output from System</u>

Output from the system is not very sophisticated currently. When any kind of remediation is returned from the RDC libraries, a textual message is reported on the PC running the IST application. This message could be sent back to the IST FITT system using a Message PDU, which in turn could play some kind of warning audio message (this would not strictly follow the DIS Standard, but would work for this limited use).

4. Conclusions

Over the past several years, the Army Research Institute Virtual Environment Research Testbed has yielded a wealth of research information on the development of complex virtual environments. This report describes the work performed for integrating in the RDC tutoring system libraries into an application that can co-exist on the DIS network with the IST FITT system. IST chose to implement a general-purpose solution to maximize future possibilities with the system and provide a flexible mechanism for changes. This is particularly important for any bug fixes that may be needed in the future especially regarding any misunderstandings of the RDC libraries (due to limited documentation). However, the IST application developed is, in the very least, a good foundation for integrating the two systems together.