RL-TR-96-258
Final Technical Report
March 1997

IVIEW/DIS INTEGRATION

Amherst Systems, Inc.

Paul Castiglione and Kristen Staub

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-96-258 has been reviewed and is approved for publication.

APPROVED:

ALEX F. SISTI
Project Engineer

FOR THE COMMANDER:

JOSEPH CAMERA, Technical Director
Intelligence & Reconnaissance Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1997

3. REPORT TYPE AND DATES COVERED
Final Sep 95 - Aug 96

4. TITLE AND SUBTITLE
IVIEW/DIS INTEGRATION

5. FUNDING NUMBERS
C: F30602-95-C-0233
PE: 62702F
PR: 4594
TA: 15
WU: N2

6. AUTHOR(S)
Paul Castiglione and Kristen Staub

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Amherst Systems, Inc.
30 Wilson Road
Buffalo NY 14221-7082

8. PERFORMING ORGANIZATION REPORT NUMBER
704-9160005

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory/IRAE
32 Hangar Road
Rome NY 13441-4114

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-96-258

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Alex F. Sisti/IRAE/(315) 330-4518

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report documents the design and development of a visualization tool for use within both the Distributed Interactive Simulation (DIS) and the Modeling and Simulation communities. This software is based on the IVIEW 2000 visualizer, which was developed for the National Air Intelligence Center (NAIC) to provide a visualization environment for analysis of air-to-air engagements. Combining this package with a DIS interface provides analysts and developers with a free government-owned visualization tool.

14. SUBJECT TERMS
Distributed Interactive Simulation (DIS)
IVIEW 2000
Visualization

15. NUMBER OF PAGES
24

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
Same as Report

TABLE OF CONTENTS

SECTION PAGE

Section 1
SCOPE

1.1 Identification

This document is the Final Report for the Distributed Interactive Simulation (DIS) interface software to the IVIEW 2000 developed for contract number F30602-95-C-0233, IVIEW/DIS Integration.

1.2 Problem Description

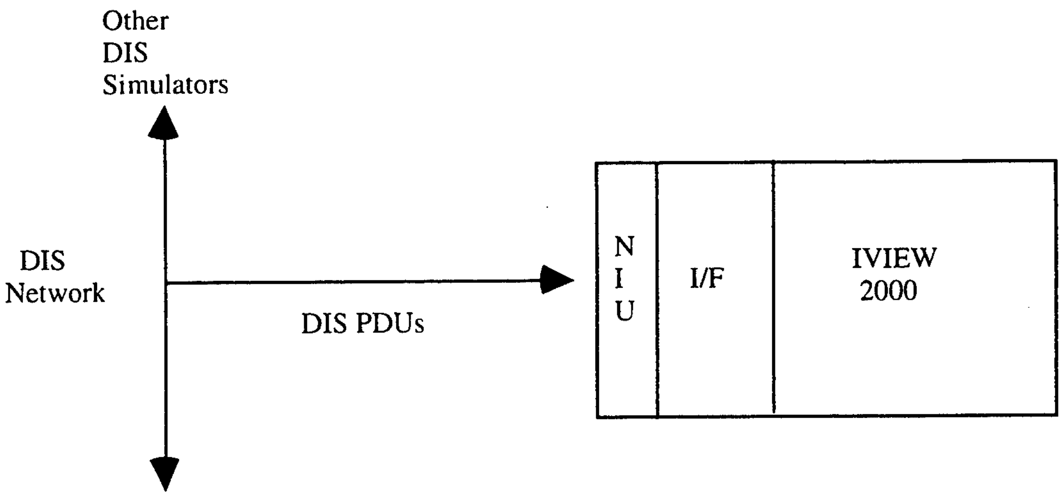
As part of Rome Lab's Modeling and Simulation efforts, it has sponsored the development of a government owned tool for visualizing DIS exercises. IVIEW 2000 is the basis for this new visualization tool.

The Air Force started with a 3-dimensional visualization tool that it owned, IVIEW 4.1. This was a non-real-time visualization tool which read in data from a time-based flight history file. Many features have since been enhanced or added to this tool, which has been upgraded to IVIEW 2000. The input data can describe the flight characteristics of air-to-air engagements, or data from other model simulations, such as TAC Brawler (an air-to-air engagement model) or Trap (a missile model). The scenario data can be played and reviewed, scaled and seen from several different points of reference. This provides a very useful tool for analysts to view engagements.

Rome Laboratory sponsored the effort to attach a DIS interface to this visualization tool. This enhancement now allows IVIEW users to participate in DIS exercises. Predefined data is no longer required for IVIEW 2000 users. Dynamic DIS player data can now be operated on by IVIEW 2000 via the DIS interface. To participate in a DIS exercise, a hard requirement for the system is real-time operation. The IVIEW/DIS interface also added the required real-time processing to IVIEW 2000, enabling it to realistically display the participants in DIS exercises.

1.3 System Overview

Amherst Systems has developed software that enables IVIEW 2000 to execute in real-time and receive information from remote simulators using Distributed Interactive Simulation (DIS) messages. These messages, called Protocol Data Units (PDUs), contain "ground truth" information. The newly developed software processes the DIS messages and gives the necessary information to IVIEW 2000, so that DIS players are displayed through IVIEW 2000. Figure 1.3-1 illustrates the architecture.



A DIS Network Interface Unit (NIU) has been added to IVIEW 2000 providing communication with a DIS Network. This NIU executes as a receive only node, and also performs PDU verification and filter processing based on DIS version, PDU type, and DIS exercise ID.

With the addition of the DIS capability to IVIEW 2000 comes the ability to perform real-time processing. Modifications were made to the IVIEW 2000 software to accommodate real-time execution. Without the DIS interface, IVIEW 2000 processes and displays the preset player flight path data contained in its input file. The DIS Network Interface Unit replaces this preset data, and now reads DIS Entity State PDUs in real-time, obtains the necessary player position data information, and passes the player data to IVIEW 2000 for updating the displays.

Processing exists to properly unpack and process the player data contained in the DIS Entity State PDU. The PDU is defined using the DIS standard 2.0.4. The positional data is used to update the IVIEW 2000 displays. The entity type data is obtained from the PDU and mapped to corresponding icons which were created by IVIEW 2000. The IVIEW 2000 data structures are updated with the new DIS information.

DIS Player processing performs the required DIS functions of timing out external players after not receiving an Entity State PDU for the specified period of time. In addition, there is processing to perform the dead reckoning of DIS players to predict their future location. The DIS software also interfaces the required player data to properly update the IVIEW 2000 data structures required for display updates.

1.4 Document Overview

This document addresses the effort to implement the DIS standard specified in the document "Standard For Interactive Simulation Protocols For Distributed Interactive Simulation Applications, Version 2.0 (Fourth Draft)". Some key capabilities that were implemented include the ability to communicate using the DIS communication architecture, the proper implementation of DIS algorithms, the ability to receive Entity State PDUs, and the ability to display the remotely located simulated players within IVIEW 2000.

Section 2
REFERENCED DOCUMENTS

Enumeration and Bit-Encoded Values for use with IEEE 1278.1-1995, Standard for Distributed Interactive Simulation--Application Protocols, IST-CR-95-14, prepared by Institute for Simulation and Training

IVIEW 2000 Project User's Manual Version 1.0, April 17, 1995, prepared by General Research Corporation

IVIEW 2000 Project, Programmer's Guide, Version 1.0, April 17, 1995, prepared by General Research Corporation

Standard For Interactive Simulation Protocols For Distributed Interactive Simulation Applications, Version 2.0 (Fourth Draft), prepared by Institute for Simulation and Training

Section 3
IMPLEMENTATION OVERVIEW

Necessary changes to the IVIEW 2000 software were incorporated to allow the activation of the DIS Interface feature. Risk was reduced by modifying only a small number of IVIEW 2000 routines. Also, the data interface between the two programs was minimized. This design approach helps future modifications of either IVIEW 2000 or the DIS Interface software to be upgraded independently, while minimizing side-effects.

3.1 File: ScenarioMgr.c

3.1.1 Function: SM_Create

This function was modified to call the DIS software routine, DISMGR_Initialize, as part of the IVIEW 2000 startup initialization processing. This DIS routine reads in data from the DIS Configuration file and performs all the initialization processing required by the DIS software.

3.1.2 Function: SM_Destroy

This function was modified to call the DIS software routine, DISMGR_Cleanup, as part of the IVIEW 2000 shutdown processing. This DIS routine closes any sockets that are in use. DISMGR_Cleanup also writes the DIS counters to the DIS log file.

3.1.3 Function: SM_GetIcon

IVIEW 2000 creates data structures for all icons that are specified in the AFDR file. Only those icons identified in this file can be used. This function was modified to enable the DIS software to gain access to the newly created IVIEW icon. This information is required for mapping DIS players to IVIEW 2000 icons. After IVIEW 2000 performs its normal processing to create the icon contained in the AFDR file, that data is stored in its data structures. The DIS software is then called to obtain proper pointers to this new icon data.

3.1.4 Function: SM_LoadFDRFile

IVIEW 2000 creates data structures for all icons that are specified in the AFDR file. Only those icons identified in this file can be used. This function was modified to enable the DIS software to read in new IVIEW 2000 icons. This information is required for mapping DIS players to IVIEW 2000 icons. First, an initialization is performed to reset DIS icon mapping data to allow an AFDR file to be reloaded. Next, IVIEW 2000 performs its normal processing to read in the data contained in the AFDR file. Finally, the DIS software is again called to perform verification of the icon data that was read in.

3.1.5 Function: SM_UpdateCurrentTime

This function was modified to read DIS player data and insert it into the IVIEW data structures. This allows the IVIEW 2000 display software to update the Viewing Display with data that corresponds to DIS players.

3.2 File: TimeSynch.c

3.2.1 Function: TS_workProcCallback

This function was modified to call the DIS software routine, NIU_Executive. This DIS routine controls the interface processing required for receiving Protocol Data Units (PDUs) from the Ethernet board.

Before the DIS interface to IVIEW 2000 was added, IVIEW 2000 used 100% of the CPU. A single call from TS_workProcCallback allows the DIS software to gain access to the CPU without affecting the overall structure of IVIEW 2000.

3.3 File: TimeSynch.h and TimeSynch.c

3.3.1 Function: InitializeSeconds

The function InitializeSeconds was not modified in any way. The files were modified only to allow the DIS software visibility to it. This was required to allow IVIEW 2000 and DIS to be using the same time frame. The function prototype was moved from being defined in the TimeSynch.c file to the header file TimeSynch.h.

3.3.2 Function: GetSystemTime

The function GetSystemTime was not modified in any way. The files were modified only to allow the DIS software visibility to it. This was required to allow IVIEW 2000 and DIS to be using the same time frame. The function prototype was moved from being defined in the TimeSynch.c file to the header file TimeSynch.h.

3.4 File: MainWinCBs.c

3.4.1 Function: MW_ExitCB

The function MW_ExitCB was modified to call the IVIEW 2000 Scenario Manager function SM_Destroy. This is required to allow the DIS software to perform an orderly shutdown.

Section 4
PROBLEMS ENCOUNTERED/ LESSONS LEARNED

This section identifies problems that were encountered or lessons learned during the IVIEW/DIS Integration effort.

We were able to add in many more features into this development effort due to the fact that we had already performed the engineering phase of determining how a DIS NIU should perform. Additional savings were realized in the DIS player processing portion of the IVIEW/DIS program. Here again many higher level features were added because of our previous DIS experience.

We learned some valuable lessons about software re-usability. This program made use of a large amount of software, which needed to be re-hosted instead of developed. The majority of the DIS NIU and Player software was re-hosted from Ada to C. Minor adjustments to the software were made to preserve the packaging of the objects.

There were problems inherent in the development of software required to interface with another unreleased version of a product. Although there was no way to avoid this, it did have an effect on the ongoing development effort. As the formal release of IVIEW 2000 was delayed, so too was this effort; IVIEW/DIS was forced to halt work to minimize any cost impact. There were dramatic changes in the icon processing between IVIEW 2000 versions to upgrade the icon processing/capabilities. As a result of this icon data interface changing, certain modifications needed to be made to the DIS software, which directly interfaces with this data to perform DIS player mapping.

The IVIEW 2000 developers, General Research Corporation, did a good job in both the development of their software and its support. Their implementation of object-oriented C software was done very well, which eased the learning curve associated with the interface. Also, they provided us with good support when we did have questions or problems concerning the interface or the changes that were made.

Section 5
CONCLUSION

5.1 Enhancements

The following is a list of enhancements that could be added to the DIS side of the IVIEW/DIS program.

Process additional PDUs

> The Fire, Detonation, and Collision PDUs can easily be added. The incoming PDU processing must be updated at a minimum. However, to give additional visual effects, various explosion or fire/flash icons can be used to indicate that these events have occurred.

Log the dynamic DIS data to a file and process it to create a standard IVIEW 2000 AFDR file

> The DIS data as it has been received and dead reckoned, can be written to an ASCII file. This data can be later processed by another small utility that would transform this data to an IVIEW2000 AFDR File. This would then allow a valuable playback utility for analysts. The true dynamic DIS data can be captured and then transformed into a standard AFDR file for later use and re-use. This is a very valuable feature.

Perform Dead Reckoning on orientation

> To provide a higher fidelity model, the dead reckoning processing should be expanded to incorporate orientation. This processing is more computationally intensive which can dramatically effort performance, but provisions can be incorporated to minimize this processing. This feature is not warranted until the coordinate system issue has been resolved. Once IVIEW 2000 supports a spherical coordinate systems, particularly WGS-84 using ECEF, then the orientation processing should be implemented.

Controlling the Viewing Window update rate

> One method to accommodate this is to add features to the VCR Control Panel to easily adjust the display update rate using + and - keys. This will provide the user with control to fine tune the visual feedback of the scenario.
>
> Another method of controlling the screen update rate would be to add processing to the software to measure, determine, and adjust the display update

rate dynamically. This would automate this feature for the user, but provide a slight additional processing overhead.

Add user protection features to VCD Control Function

Protection features can easily be added that lock out the user from selecting either Step Advance and Rewind keys when running executing in a DIS scenario. Currently, this is left as an instruction to the user to not to press these keys. Other similar user protection features could also be added.

Incorporate 3-D Trails for real-time DIS exercise

Longer 3-D Trails could be made available to the user by adding additional memory for history data. This could implemented if desired by the user community. It should be noted that if the above mentioned enhancement of the playback utility is implemented, the standard IVIEW2000 3-D trail displays will be automatically provided at this point.

Re-host NIU software to execute on a separate CPU

During the development of the IVIEW/DIS program, consideration was made to ease the porting of this program to a multiple processing system. This can be performed if IVIEW/DIS is to participate in more dense real-time DIS exercises. At this point, the NIU software should be placed on a separate processor to accommodate heavy network traffic.

Addition of On-Line Help Feature for IVIEW/DIS operation

An On-Line Help Feature would definitely increase these ease of use for the IVIEW/DIS program. It could easily be incorporated to appear like a seamless extension of the IVIEW 2000 Help Feature. The majority of the Software Users Manual could be implemented on-line as an aide to the users.

Update of DIS versions

As the DIS standards continue to evolve, the IVIEW/DIS software should likewise be updated to accommodate these new standards. Provisions were already incorporated in the design of IVIEW/DIS to allow for this anticipated update with ease.

5.2 Follow On Possibilities

The following is a list of enhancements that Amherst Systems could support or provide to the entire IVIEW/DIS program.

- Support of WGS-84 world map using ECEF coordinates for IVIEW/DIS

- Support of Terrain processing for IVIEW/DIS
- Support for multi-lab testing of Simulation and Modeling exercises
- Support for development of terrain icon
- Enhancements for icon rendering

5.3 Papers on IVIEW/DIS

Amherst Systems submitted a paper distributed at the DIS 15th Workshop meeting in Orlando, FL. The paper, titled *"The Development of a Government-Owned DIS Visualization Tool"*, describes the IVIEW/DIS program to the DIS community. It appears in Appendix A of this report.

Appendix A
IVIEW/DIS PAPER

THE DEVELOPMENT OF A GOVERNMENT-OWNED DIS VISUALIZATION TOOL

Paul Castiglione
Amherst Systems, Inc.
30 Wilson Road
Buffalo, NY 14221
(716) 631 - 0610
castiglione@amherst.com

Keywords: Visualization IVIEW IVIEW2000

ABSTRACT

Amherst Systems, Inc., under the sponsorship of Rome Laboratory, has developed a visualization tool for use within both the DIS and the Simulation and Modeling communities. This product uses the IVIEW2000 software provided by the Air Force's National Air Intelligence Center (NAIC). The capabilities of IVIEW2000, combined with a DIS interface, provide analysts and developers with a free, government-owned visualization tool for the evaluation of DIS exercises.

IVIEW2000 is a software package providing a visualization environment for engagement reconstruction used by the simulation, modeling and analysis community. IVIEW2000 provides many different viewing capabilities, access to player data, and a fully featured control panel with VCR-like functions. It allows analysts to view the entire or selected portions of the scenario from a variety of different angles simultaneously, while monitoring various types of player data. Prior to the addition of the DIS Interface, IVIEW2000 allowed only scripted scenarios to be played and re-played.

With the addition of the DIS Interface, IVIEW2000 is now able to read and process DIS PDUs and operate as a real-time visualization tool. It allows an operator to attach to any/multiple DIS entity(s), provides bird's-eye view, out-the-window views, data-readouts, etc.

IVIEW2000, with the DIS Interface, operates on a standard SGI Indigo-class machine, making it an affordable alternative to commercial DIS visualization packages.

INTRODUCTION

As part of Rome Lab's Modeling and Simulation efforts, it has sponsored the development of a government owned tool for visualizing DIS exercises. IVIEW 2000 is the basis for this new visualization tool.

The Air Force started with a 3-dimensional visualization tool that it owned, IVIEW 4.1. This was a non-real-time visualization tool which read in data from a time-based flight history file. Many features have since been enhanced or added to this tool, which has been upgraded to IVIEW 2000. The input data can describe the flight characteristics of air-to-air engagements, or data from other model simulations such as TAC Brawler (an air-to-air engagement model), or Trap (a missile model). The scenario data can be played and reviewed, scaled and seen from several different points of reference. This provides a very useful tool for analysts to view engagements.

Recently Rome Laboratory sponsored the effort to attach a DIS interface to this visualization tool. This enhancement now allows IVIEW users to participate in DIS exercises. Predefined data is no longer required for IVIEW 2000 users. Dynamic DIS player data can now be operated on by IVIEW 2000 via the DIS interface. To participate in a DIS exercise, a hard requirement for the system is real-time operation. The IVIEW/DIS interface also added the required real-time processing to IVIEW 2000, enabling it to realistically display the DIS participants in DIS exercises.

IVIEW 2000

Background

Rome Lab's Intelligence and Reconnaissance Branch (RL/IRAE) has long provided modeling and simulation support to a variety of intelligence consumers and analysts across a wide spectrum of application areas. This support includes basic and applied Research and Development in the enabling technologies of Modeling and Simulation, proof-of-concept and feasibility prototype development efforts, system and component design supports, and ultimate transition to field exercises and operational sites. Through Rome Lab's involvement in these areas, the visualization tool, IVIEW 2000 has evolved.

Purpose

IVIEW 2000 is a software package providing a visualization environment for engagement reconstruction used by the simulation, modeling and analysis community. It was initially developed at the National Air Intelligence Center (NAIC), to support NAIC/TAAE analysts. Compared to other tools, IVIEW 2000 provides the users with much greater flexibility in terms of platform independence, configuring and saving engagement scenarios, and visualizing various aspects of simulated engagements. Providing a graphical replay capability, IVIEW 2000 meets the needs of the modeling, simulation and analysis community.

Description

IVIEW 2000 is a dynamic, modular, flexible software package, developed by the Air Force (Rome Laboratory), with capabilities for real-time 3-dimensional animation characteristics. It displays time-based history files of missile and aircraft trajectories. The graphical presentation package allows 3-dimensional visualizations of player activity. IVIEW 2000 has many applications such as supporting analysis of aerodynamic systems engagements and supporting dynamic weapon platform analysis models. Visualizing models for ballistic missiles and space systems is another research application IVIEW 2000 supports.

The input data can be generated by one of several different non-real-time analysis models. Example of these models include TAC Brawler and Trap. IVIEW 2000 processes the models data to view aircraft and missile flight activity over flat or real terrain.

The IVIEW 2000 system has been delivered by the contractor General Research Corporation (GRC) of Dayton, OH. It executes on either a Silicon Graphics or a Sun workstation, generally to post-process a variety of engagement simulations. It operates from a processed input file, which identifies each players' type, team, and movement in 3-dimensions. This scenario data can be played and re-played both in forward or reverse directions, allowing the user greater flexibility in controlling the scenario. The display can be adjusted to be viewed from above, from the cockpit, or from a wing-man's perspective, giving a graphic presentation of the multiple engagements.

Features

IVIEW 2000 provides many different viewing capabilities, access to player data, and a fully featured control panel with VCR-like functions. It allows analysts to view the entire or selected portions of the scenario from a variety of different angles simultaneously, while also monitoring various types of player data.

Scenario execution is easily controlled by a VCR-like control panel. The control panel allows the scenario to start/stop/freeze upon selection. Additional controls enable the user to continue execution by either frame-by-frame advancement or at clock speed in either a forward or backward direction. This allows the user full control over scenario execution.

Scenario players are displayed to the user through the Viewing Window. Multiple viewing windows may be enabled simultaneously allowing the user visual inspection from both perspectives of two engaging aircraft. Additional options allow for player labels or numbers to be displayed upon the corresponding icon to aide in the identification of players. Each Viewing Window supports several viewing angles: birds-eye view, pilot's-view with an optional Heads-Up-Display (HUD), and a trail-behind view. In addition to these different viewing perspectives, the user can view the engagement reconstruction from any point in 3-Dimensional space. The icon objects displayed within these windows can be scaled to adjust their magnification to enhance visibility of distant players. These controls are presented to the user in an easy to use graphical user interface. Through the use of the mouse, the user activates commands which take effect immediately.

The Data Window can be enabled to output current positional information with respect to each player in the scenario. This window allows the user to view a player's positional information, in numeric format, while the scenario is executing. Data is constantly updated as the player follows its flight path through the scenario.

The Message Data Window displays message information that is communicated between two players. This message information includes the message itself, identification of the sender and receiver of the message, along with its timestamp.

The Sensor Data window is a two-dimensional display indicating information from the scenario file in a user-definable graphical or numerical format. Infrared Search and Track(IRSTs) and radars are examples of sensors that can be viewed using this window.

To compare the positions or altitudes of different players, a terrain grid can be enabled on the Viewing Window. The terrain grid can be displayed both vertical and/or horizontal to help create a plane of reference.

Screen configurations can be saved to permit quicker and easier setup time between scenarios. This is a very useful and helpful feature, allowing ten different users their own personal setup configuration. Setup configurations are easily loaded and/or saved at anytime during program execution.

On-Line Help, using context sensitive hyper-text, is also available to the user. This provides a very efficient way to quickly support user questions. Help can be activated using either a Context based selection or an Indexed listing.

IVIEW 2000 supports input from other model simulations such as TAC Brawler (an air-to-air engagement model) or Trap (a missile model). Input files generated from these simulations are able to be processed by IVIEW 2000, which can provide a good means to visually display the actual engagement scenario.

NEW DIS CAPABILITY TO IVIEW 2000

Purpose

The purpose of adding a DIS interface to IVIEW 2000 is to develop a capability for the Air Force to use their graphical visualization tool, IVIEW 2000, to monitor Distributed Interactive Simulation (DIS) exercises. The resultant system provides DIS compatibility to the current IVIEW community, and a free, non-proprietary visualization tool for the DIS community, accommodating both user groups with much needed features.

Additionally, for those users who already have a DIS compliant simulator, the IVIEW/DIS program provides a useful visualization tool of the entire DIS scenario, capable of simultaneously viewing multiple players from various perspectives. For those users who are currently building a DIS compliant simulator, the IVIEW/DIS program provides a means to verify many DIS features and algorithms. The simulator's position and orientation equations, its rotation and translation processing to earth-centered-earth-fixed (ECEF) coordinates, as well as its Dead Reckon algorithms are just a few examples of the processing where IVIEW/DIS can provide an important visual feedback of how other DIS simulators are "seeing" the simulated entity.

Description

A DIS Network Interface Unit (NIU) has been added to IVIEW 2000 providing communication with a DIS Network. This NIU executes as a receive only node, and also performs PDU verification and filter processing based on DIS version, PDU type, and DIS exercise ID.

With the addition of the DIS capability to IVIEW 2000, comes the ability to perform real-time processing. Modifications were made to the IVIEW 2000 software to accommodate real-time execution. Without the DIS interface, IVIEW 2000 processes and displays the preset player flight path data contained in its input file. The DIS Network Interface Unit, replaces this preset data, and now reads DIS Entity State PDUs in real-time, obtains the necessary player position data information, and passes the player data to IVIEW 2000 for updating the displays.

Processing exists to properly unpack and process the player data contained in the DIS Entity State PDU. The PDU is defined using the DIS standard 2.0.4. The positional data is used to update the IVIEW 2000 displays. The entity type data is obtained from the PDU and mapped to corresponding icons which were created by IVIEW 2000. The IVIEW 2000 data structures are updated with the new DIS information.

DIS Player processing performs the required DIS functions of timing out external players after not receiving an Entity State PDU for the specified period of time. In addition, there is processing to perform the dead reckoning of DIS players to predict their future location. The DIS software also interfaces the required player data to properly update the IVIEW 2000 data structures required for display updates.

Features

A configuration files exists to ease the initialization procedure for starting a DIS exercise. The user can specify in this configuration file the values for DIS data that may change from exercise to exercise (e.g. DIS Exercise ID). Additional features specified through this configuration file are the enabling/disabling of filter processing as well as identifying player entity types and their associated icons. This data is read from an ASCII file and does not require the re-compilation of source code to accommodate these changes.

A log file is output after a DIS exercise is complete. Data contained here is a description of the input startup parameters, performance monitoring data, along with any error messages that may have been encountered during program execution.

With the constant evolution of the DIS standard, IVIEW/DIS has been developed to accommodate multiple versions of DIS. This will not only prolong the life of the product, but also allow easier upgradeability as new DIS version standards are developed. It will also provide a interim compatibility as simulators are being upgraded to accommodate the new released DIS version. The DIS version in use is specified in the above described configuration file.

For many exercises, not all players are of vital importance. For example, if one is only concerned with analyzing an air-to-air engagement occurring at 30,000 feet, dismounted infantry entities may not be all that significant. If too much time and effort is spent processing and drawing insignificant players, this processing overhead could cause a degradation in performance. The IVIEW/DIS program addresses these issues by providing the ability to filter out DIS players based on their Entity Type as indicated in the Entity State PDU. This feature should be used with caution, but can provide a drastic improvement in throughput and display clarity.

The initialization and correlation of entity types to icons is performed at startup. The configuration file should contain all of the entity types that will be used during the DIS exercise. The definition of the Entity Type along with its corresponding icon are simply listed in the file, which is parsed at startup. If the DIS enumerations that identify the entity type are changed or updated, then only the configuration file must be edited and updated. This feature allows for this type of flexibility by requiring no software changes when DIS enumerations are updated.

Default icons for certain categories of entities can be specified. This solves a problem with not being able to see an external player because that site does not have the correct version of the enumerations for the DIS version in use. For example, let's assume the entity type definition of an F-15E changed between DIS versions. If one F-15E simulator in the exercise did not make the appropriate changes, that F-15E may not be seen/accepted/validated to all others simulators in the exercise. However, provisions have been made within IVIEW/DIS to accommodate this and allow that player to be seen / mapped to a default icon. This default icon for the air platform, which may be a generic aircraft icon, will indicate that a player in the exercise is not using the identical enumerations as the rest of the exercise, but it will allow it to be seen or displayed on the IVIEW 2000 displays.

Another use of this feature is that it can minimize the startup coordination of a DIS exercise by ensuring all participants have the identical Entity Type descriptions for each player. This feature allows IVIEW/DIS to start executing at any point during an ongoing DIS exercise. IVIEW/DIS is not limited by the requirement of knowing every Entity Type description of each participant prior to the start of a DIS exercise.

The exact description or definition of each new DIS player is logged to an output file. This feature allows IVIEW/DIS to join an ongoing DIS exercise and record all the players entity type descriptions to a file. After running long enough to receive at least one Entity State PDU from each player, the IVIEW/DIS program can be stopped. If it is determined that the icon mapping using the default icons for any player is not acceptable, the output log file can be examined to determined the individual entity type fields. Then, the input configuration file can be updated to specifically account for each entity type in the scenario. Upon restarting IVIEW/DIS and re-joining the DIS exercise, all players will be correctly identified and mapped to their correct icon.

Dead Reckoning is performed on the valid DIS players to compute the estimated location of these entities between receipt of Entity State PDUs. This processing estimates the location of each dynamic DIS player between PDU updates. This data is passed to the IVIEW 2000 software to allow its drawing routines access to the correct information on where to place the icon within the display.

CAPABILITY OF IVIEW/DIS

A need exists in the modeling and simulation community for a government owned visualization tool to aid in the development and implementation of DIS compliant software. This type of software greatly enhances modeling and simulation for supporting the intelligence mission. By integrating DIS with the power of a graphics animation, DIS implementers are able to immediately obtain visual correlation of simulator performance. The resulting system is a tool that provides for animation of all DIS entities participating in a DIS exercise.

Prior to the addition of the DIS Interface, IVIEW 2000 allowed only scripted scenarios with preset data to be played and re-played in a non-real-time environment. Although IVIEW 2000 can process input data from some other models (TAC Brawler and Trap), it is very tedious and time consuming effort to create a scripted scenario input file.

With the addition of the DIS Interface, IVIEW 2000 is now able to read and process DIS PDUs and operate as a real-time visualization tool. It allows an operator to attach to any/multiple DIS entity(s), providing viewing from many different perspectives, obtaining numerical data readouts, and supplying developers with a means of evaluating and testing DIS algorithms.

Within the Modeling and Simulation community, there are many simulations, models, and databases that act as stand alone systems. The use of DIS technology within this environment provides the means for these systems to interact and create a setting that can be used for test and evaluation of Command, Control, Communications and Intelligence (C^3I) systems and procedures. This type of integration allows testing between geographically dispersed DIS sites. Interconnecting legacy models or simulations can create a synthetic battlefield.

There is minimal user input or setup required to allow the IVIEW/DIS program to participate in a DIS exercise. Accommodations have been made to ease the user's effort of inputting the required configurable data for DIS exercises.

The design approach minimizes the user's effort for updating to future DIS releases and IVIEW releases. IVIEW/DIS can be very easily modified to accommodate the processing of additional PDUs in the future. IVIEW/DIS has also been designed flexible enough to support multiple versions of DIS within the same executable, providing a more durable, practical and adaptable product.

IVIEW/DIS provides owners of DIS simulators with a visualization tool. IVIEW/DIS provides new developers of DIS simulators not only with a visualization tool, but also with visual aides in creating DIS simulators, testing, evaluating DIS Dead Reckoning algorithms.

REFERENCES

Rome Laboratory's Web Site

IVIEW 2000 User's Guide

IVIEW 2000 On-Line Help

☆U.S. GOVERNMENT PRINTING OFFICE: 1997-509-127-47157

MISSION
OF
ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.