1-1-1995

Implementation Of IST's TRIDIS Stealth

William K. Andrews

Find similar works at: https://stars.library.ucf.edu/istlibrary
University of Central Florida Libraries http://library.ucf.edu

This Research Report is brought to you for free and open access by the Digital Collections at STARS. It has been accepted for inclusion in Institute for Simulation and Training by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

Recommended Citation
https://stars.library.ucf.edu/istlibrary/113
Implementation of IST's TRIDIS Stealth

Final Report

Institute for Simulation and Training
3280 Progress Drive
Crawford FL 32826

University of Central Florida
Division of Sponsored Research
# TABLE OF CONTENTS

1.0 Introduction .......................................................................................................................... 1  
  1.1 Acronyms ............................................................................................................................... 1  
  1.2 Purpose .................................................................................................................................. 2  
  1.3 Background ............................................................................................................................. 2  
  1.4 Contents of Report .................................................................................................................. 3  

2.4 Task Description ....................................................................................................................... 3  
  2.4.1 Task History ...................................................................................................................... 3  
  2.4.2 Description of Stealth System .......................................................................................... 4  
  2.4.3 Design Philosophy ............................................................................................................ 4

3.0 Requirements Analysis ............................................................................................................. 5  
  3.1 Visual System .......................................................................................................................... 5  
    3.1.1 Visual Hardware and Driver Support .............................................................................. 5  
      3.1.1.1 Texture ......................................................................................................................... 5  
      3.1.1.2 Shading ........................................................................................................................ 5  
      3.1.1.3 Visual Resolution ....................................................................................................... 6  
    3.1.2 Visual Data ......................................................................................................................... 6  
      3.1.2.1 Visual Terrain ............................................................................................................. 6  
      3.1.2.2 Visual Models ............................................................................................................. 7  
      3.1.2.3 Environmental Effects ............................................................................................... 7  
    3.1.3 Update Rate ...................................................................................................................... 7  
    3.1.4 Image Capacity ................................................................................................................. 7  
  3.2 Interface Requirements ........................................................................................................... 8  
    3.2.1 User Interface ................................................................................................................... 8  
    3.2.2 DIS Interface .................................................................................................................... 8  
      3.2.2.1 Network Capacity ....................................................................................................... 9  
      3.2.2.2 Entity Capacity ........................................................................................................... 9  
    3.2.3 Interface Hardware .......................................................................................................... 9  
      3.2.3.1 VME ............................................................................................................................ 9  
      3.2.3.2 EtherNet ..................................................................................................................... 10  
      3.2.3.3 FDDI .......................................................................................................................... 10  
    3.2.4 Visual Interface Manager ................................................................................................. 10

4.0 Product Survey ......................................................................................................................... 11  
  4.1 Hardware Platforms ............................................................................................................... 11  
  4.2 Software Products .................................................................................................................. 11  
    4.2.1 Visual Systems ................................................................................................................ 11
4.2.1.1 Silicon Graphics Performer .................................................. 12
  4.2.1.1.1 Coryphaeus EasyScene .................................................. 12
  4.2.1.1.2 Paradigm ............................................................... 12
    4.2.1.1.3 Mak Technologies VR - Stealth ..................................... 12
  4.2.1.2 GL/Open GL .............................................................. 13
    4.2.1.2.1 Gemini Technology .................................................. 13
    4.2.1.2.2 Computer Explorations Inc. ....................................... 13
  4.2.1.3 Loral 2D/3D Viewer ..................................................... 13
  4.2.2 Visual Modelers ............................................................ 14
    4.2.2.1 Designers Workbench - Coryphaeus .................................... 14
    4.2.2.2 MultiGen .................................................................. 14
    4.2.2.3 S1000 - Loral/BBN ..................................................... 15
  4.2.3 Terrain Modelers .................................................................. 15
    4.2.3.1 EasyT - Coryphaeus ...................................................... 15
    4.2.3.2 MultiGen ................................................................... 15
  4.2.4 DIS Interfaces ..................................................................... 16
    4.2.4.1 DIP (NAWC/TSD Aladdin) ............................................... 16
    4.2.4.2 Mak Technologies - VR-Link .......................................... 16
    4.2.4.3 NRaD SGI Application Interface ..................................... 16
    4.2.4.4 Loral Interface Library Developers Kit (ILD) ...................... 17

5.0 Final Selections ........................................................................ 17
  5.1 Visual System ........................................................................ 17
  5.2 Visual Modeler ...................................................................... 18
    5.2.1 Visual Models .................................................................. 18
  5.3 Terrain Modeler .................................................................... 18
  5.4 DIS Interface ........................................................................ 19
  5.5 User Interface ....................................................................... 19

6.0 Development Issues .................................................................. 19
  6.1 Visual System ........................................................................ 19
    6.1.1 Visual Model Loading ....................................................... 19
    6.1.2 Terrain Loading .............................................................. 20
    6.1.3 Terrain Databases ............................................................ 20
    6.1.4 Ground Clamping ............................................................. 22
  6.2 DIS Interfaces ........................................................................ 22
    6.2.1 Stealth PDU Experiments .................................................. 23
    6.2.2 Stealth PDU Utilization for I/ITSEC '94 ............................... 23
      6.2.2.1 Experimental Stealth PDU .............................................. 23
        6.2.2.1.1 PDU Header ............................................................ 23
        6.2.2.1.2 Force ID ................................................................. 23
        6.2.2.1.3 Entity Type ............................................................ 23
        6.2.2.1.4 Entity Position and Velocity .................................... 24
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.2.1.5 Entity Orientation</td>
<td>24</td>
</tr>
<tr>
<td>6.2.2.2 I/ITSEC Experimental Stealth PDU Problems and Recommendations</td>
<td>24</td>
</tr>
<tr>
<td>6.2.2.2.1 Latency Mismatch Jitter</td>
<td>24</td>
</tr>
<tr>
<td>6.2.2.2.2 Offset Stacking</td>
<td>25</td>
</tr>
<tr>
<td>6.2.3 The Stealth Control PDU</td>
<td>26</td>
</tr>
<tr>
<td>6.2.3.1 Conceptual Model for Stealth Control</td>
<td>26</td>
</tr>
<tr>
<td>6.2.3.2 Information contained in the Stealth Control PDU</td>
<td>30</td>
</tr>
<tr>
<td>7.0 Current Stealth Capabilities</td>
<td>31</td>
</tr>
<tr>
<td>7.1 Visual Scene Control</td>
<td>31</td>
</tr>
<tr>
<td>7.2 DIS Visual Appearance Support</td>
<td>32</td>
</tr>
<tr>
<td>7.3 DIS Articulation Support</td>
<td>32</td>
</tr>
<tr>
<td>7.4 Stealth Remote Control</td>
<td>32</td>
</tr>
<tr>
<td>7.5 Support for Audio System</td>
<td>32</td>
</tr>
<tr>
<td>8.0 Anticipated/Desired Enhancements to Stealth Package</td>
<td>33</td>
</tr>
<tr>
<td>8.1 Graphical User Interface for DIS Interface</td>
<td>33</td>
</tr>
<tr>
<td>8.2 Implementation of Remote Eyepoint Control</td>
<td>33</td>
</tr>
<tr>
<td>8.3 Migration of DIS Interface to High Speed Target Processor</td>
<td>33</td>
</tr>
<tr>
<td>8.4 Improved Visual Entity Selection Capability</td>
<td>34</td>
</tr>
<tr>
<td>Bibliography</td>
<td>35</td>
</tr>
<tr>
<td>Appendix A - Operational Instructions</td>
<td>36</td>
</tr>
<tr>
<td>A.1 DIS Interface Operations</td>
<td>36</td>
</tr>
<tr>
<td>A.1.1 System Configuration</td>
<td>36</td>
</tr>
<tr>
<td>A.1.2 Starting the DIS Interface</td>
<td>36</td>
</tr>
<tr>
<td>A.2.0 Visual System Operations</td>
<td>38</td>
</tr>
<tr>
<td>A.2.1 Configuring the EasyScene Visual Application</td>
<td>38</td>
</tr>
<tr>
<td>A.2.2 Starting the Visual System</td>
<td>38</td>
</tr>
<tr>
<td>A.2.3 Visual System User Interface</td>
<td>39</td>
</tr>
<tr>
<td>A.2.3.1 The Active Entity List</td>
<td>39</td>
</tr>
<tr>
<td>A.2.3.2 Eyepoint Storage</td>
<td>40</td>
</tr>
<tr>
<td>A.2.3.2.1 Storing an Eyepoint</td>
<td>40</td>
</tr>
<tr>
<td>A.2.3.2.3 Changing a Stored Eyepoint</td>
<td>40</td>
</tr>
<tr>
<td>A.2.3.2.4 Storing and Retrieving Eyepoint Lists</td>
<td>41</td>
</tr>
<tr>
<td>A.2.3.3 Attachment Mode Selection</td>
<td>41</td>
</tr>
<tr>
<td>A.2.3.4 View Mode Selection</td>
<td>42</td>
</tr>
<tr>
<td>A.2.3.5 Viewpoint Positioning</td>
<td>42</td>
</tr>
<tr>
<td>A.2.3.5.1 Lateral Movement</td>
<td>42</td>
</tr>
<tr>
<td>A.2.3.5.2 Elevation Control</td>
<td>43</td>
</tr>
<tr>
<td>A.2.3.5.3 Pitch Control</td>
<td>43</td>
</tr>
<tr>
<td>A.2.3.5.4 Heading Control</td>
<td>43</td>
</tr>
</tbody>
</table>
A.2.3.5.5 Sensitivity Control ................................. 44
A.2.3.5.6 StealthPDU Transmission ......................... 44
A.2.3.6 Visual Scene Controls ................................. 44
  A.2.3.6.1 Visibility Range ................................. 44
  A.2.3.6.2 Field of View/Zoom .............................. 44
  A.2.3.6.3 Time of Day .................................. 45
A.2.3.7 DIS Filtering Controls ............................... 45

Appendix B - Installation Instructions ....................... 46
  B.1.0 Stealth Installation Guide ......................... 46
  B.2.0 Model Loading .................................... 47
  B.3.0 Terrain Database Loading ............................ 48

Appendix C - Vendor Information ............................ 49
  C.1.0 Utilized Component Sources .......................... 49
  C.2.0 Other Noted Vendors ................................. 49
List of Figures
Figure 1 - Single Point Ground Clamping
Figure 2 - Latency Mismatch Jitter
Figure 3 - Example of Offset Stacking
Figure 4 - Mimic Attachment Mode
Figure 5 - Orbit Attachment Mode
Figure 6 - Wingman Attachment Mode
Figure 7 - Stealth Control PDU Description

Appendix A
Figure A1 - DIS Interface Root Menu
Figure A2 - DIS Interface Database Offset Menu
Figure A3 - Stealth Control Panel
Figure A4 - Eyepoint Storage Table
Figure A5 - Eyepoint Entry for Storage Table
Figure A6 - View Mode Selection Controls
Figure A7 - Eyepoint Positioning Controls
Figure A8 - Visual Scene Controls
Figure A9 - DIS Filtering Controls

Appendix B
Figure B1 - Directory Structure for Stealth Software
1.0 Introduction

This report describes the component selection and methodology used in the assembly of a visual display device for use in DIS testing.

This report is a deliverable item under subtask 3.2.2.3, "Display Devices," of the STRICOM contract #N61339-94-C-0024, entitled "TRIDIS: A TESTBED for Research in Distributed Interactive Simulation."

A copy of the Stealth software described in this report will be delivered separately. It should be noted that one of the principal components of this software is a commercially licensed application that will require the installation of a software key prior to operation.

1.1 Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIU</td>
<td>Application Interface Unit</td>
</tr>
<tr>
<td>API</td>
<td>Application Programmers Interface</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
</tr>
<tr>
<td>CGF</td>
<td>Computer Generated Forces</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off the Shelf</td>
</tr>
<tr>
<td>DIS</td>
<td>Distributed Interactive Simulation</td>
</tr>
<tr>
<td>DMA</td>
<td>Defense Mapping Agency</td>
</tr>
<tr>
<td>DR</td>
<td>Dead Reckoning</td>
</tr>
<tr>
<td>DWB</td>
<td>Designer Workbench (Corypheaus Software)</td>
</tr>
<tr>
<td>FDDI</td>
<td>Fiber Data Distributed Interface</td>
</tr>
<tr>
<td>FOV</td>
<td>Field of View</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>I/ITSEC</td>
<td>Interservice/Industry Training Systems and Education Conference</td>
</tr>
<tr>
<td>LOD</td>
<td>Level of Detail</td>
</tr>
<tr>
<td>NFS</td>
<td>Network File System</td>
</tr>
<tr>
<td>NIS</td>
<td>Network Information System</td>
</tr>
<tr>
<td>NRaD</td>
<td>Naval Research and Development</td>
</tr>
<tr>
<td>OTW</td>
<td>Out the Window (reference to display type)</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>SCPDU</td>
<td>Stealth Control PDU</td>
</tr>
<tr>
<td>SGI</td>
<td>Silicon Graphics Inc.</td>
</tr>
<tr>
<td>STRICOM</td>
<td>U.S. Army Simulation, Training and Instrumentation Command</td>
</tr>
<tr>
<td>TRIDIS</td>
<td>TESTBED for Research in Distributed Interactive Simulation</td>
</tr>
<tr>
<td>USGS</td>
<td>US Geological Survey</td>
</tr>
<tr>
<td>VIM</td>
<td>Visual Interface Manager</td>
</tr>
</tbody>
</table>
1.2 Purpose

As part of the DIS TESTBED improvement effort, under TRIDIS, DIS development and test resources have been designed and developed. One of these resources is a DIS compliant stealth visual display device. Stealth visual systems are similar to the visual display systems on a traditional vehicular simulator without the same controls or dynamics modules that make up the bulk of a training simulator. In this current DIS world, a stealth system is not generally visible to other systems, thus it can freely traverse the synthetic environment in a number of different ways without disturbing exercise participants.

This effort represents an attempt to develop a low cost solution to stealth visualization that is compatible with files and resources developed for higher end simulation applications. The requirements for this package focused on the needs of a DIS compliance tester, as well as the needs of a low-end laboratory simulator mockup. Investigations into this solution focused primarily on the Silicon Graphics platform due to existing resource requirements, however other solutions were investigated.

1.3 Background

One of the crucial aspects of testing DIS simulations is the visualization of the DIS environment. While two dimensional (2D) display systems, often called Planview Displays (PVDs), can provide inexpensive situational information, they are not useful in the visualization of entity spatial location and orientation with regard to the terrain. This information, along with the visual appearance and articulation parameters for a DIS entity can best be seen in a 3D stealth display system. Such a system allows for the rapid determination of how an entity is oriented with respect to the terrain and reduces the time required to validate behavior that is under testing and development.

With traditional 3D stealth suites having $200k plus price tags for hardware, the usage of these systems for DIS testing is not practical. As part of the effort to enhance the available DIS testing facilities, a stealth visualization device utilizing low-cost components was investigated and obtained.

It should be noted that during the course of this effort, significant changes in the cost, performance, and availability of the discussed configurations have occurred. While the hardware purchase price of the developed system was $80,000 at the beginning of this effort, superior capabilities are now available for $60,000 and will be available in a desktop configuration for under $50,000 in late summer 1995.

It is therefore evident that visual technology is evolving and expanding as fast as or faster than assessments can be prepared and validated by the research and development community, given the costs involved in the continuous acquisition of new technology. This report
discusses a stealth configuration whose components were selected based on costs and availabilitys as of June 1994.

1.4 Contents of Report

This report describes the effort undertaken to procure and configure a three dimensional display device that has been used in the development and testing of DIS interoperable simulation applications.

Included within the report are:

- Descriptions of initial technical requirements and the desired solutions;
- A brief survey of the available hardware and software options;
- The criteria used to guide the acquisition and development process;
- The rationale for each selected component along with unresolved concerns;
- Descriptions of each of the selected hardware and software components;
- The source of each selected component along with the government and corporate contact points required to obtain further information and copies of the relevant materials;
- Technical issues that arose during development;
- Descriptions of the TRIDIS Stealth system capabilities at the time this report was prepared;
- Recommendations for further refinements and new capabilities;

At the end of this report, user documentation, installation instructions, and a vendor list have been provided in appendices. These sections provided a basic working knowledge of how to install and operate the TRIDIS Stealth, but are not intended to replace the documentation provided by the original software vendors.

2.4 Task Description

2.4.1 Task History

The initial configuration for this task called for three, low-cost, single-channel, stealth display devices that would also be suitable for use as a three panel 'out-the-window-display' (OTW) or as three separate OTW displays. This was to be accomplished through the purchase of equipment compatible with the work performed at IST under contract N61339-93-C-0076, "Research and Development of Terrain Databases for DIS." The DIS interface and Stealth application software in use by that project was provided by Mak Technologies. In accordance with the anticipated requirements, three Silicon Graphics Indigo2 systems were bid along with DIS and Stealth licenses from Mak Technologies at a total cost of $76,605. These costs
proved to be erroneous due to overlooked component requirements such as additional memory and I/O devices. In addition, the capabilities of the low-end hardware and visual software (VR Stealth) proved to be less than what the DIS testing efforts required. In response to this situation, consultations were undertaken with the contract administrator from STRICOM. The results of these consultations led to a new configuration calling for a single system that represented a least practical cost, while still providing functionality compatible with the simulation industry. The final selection was a low-end Silicon Graphics Onyx system that cost $80,000 and a combination of government provided DIS and commercial visual software. While expending all of the budget on a single visual system, the project gained the ability to utilize applications and databases that companies being tested would be using in delivered systems.

2.4.2 Description of Stealth System

The Stealth system consists of three major components. These components are the DIS interface, the Visual Application Software, and a Visual Interface Manager (VIM) that mediates between the first two components. The DIS interface is responsible for communication on the DIS network. It operates independently of the Stealth components. All DIS entity and event information received by the DIS interface is placed in a shared memory partition which is visible to other applications. The visual application is responsible for rendering the visual scene and the icons that represent the DIS entities. It does not communicate directly with the DIS world; rather it depends on the Visual Interface Manager to instruct it as to what models are present along with any relevant visual characteristics. The Visual Application software will render a revised scene as frequently as system capacity will allow using whatever information was last provided. Should the DIS interface become inactive, the Visual Application software will continue to respond to user commands.

2.4.3 Design Philosophy

The software that is used for each of the three components was selected after a review of the needs of DIS testing personnel. In each case, utility and ease of use represented driving factors in the selection process. It was desired that visual code development be kept to a minimum through the use of COTS and government funded software where possible. To this end, military funded DIS software was combined with a commercial visual application through the use of an IST prepared interface manager. This produced a Stealth visual package that permits the visualization of most of the characteristics of DIS entities without the need to generate visual display code. Each of the reviewed products is described in the report that follows, along with descriptions and rationales for why a given product was chosen.
3.0 Requirements Analysis

During the early stages of the Stealth task, a basic set of operational requirements was defined. These requirements are limited to stealth visualization and are not intended for the configuration of a simulator’s visual display system.

3.1 Visual System

3.1.1 Visual Hardware and Driver Support

The basic required capabilities for the visual hardware and drivers are discussed below. These capabilities were based on the materials available and anticipated testing requirements for DIS [Vanzant-Hodge94]. These requirements do not reflect a complete ‘wish list’ for visual capabilities, but rather the basic requirements for the visualization of DIS entity characteristics and visual effects (i.e. detonations) that client companies are expected to have tested.

3.1.1.1 Texture

The system must provide texture processing for rendered polygons. The visual scene texturing must provide for activation and deactivation during run time.

The driving factor for this requirement is the fact that most of the visual models and effects to be utilized are generated using textured polygons so as to reduce the polygon counts in the visual scene. Without texture, it can be very difficult to differentiate between visual components. Texture generation can impose excessive demands on the computer driving the display device if the visual hardware does not handle texture internally. To allow for the use of less expensive hardware, the texture rendering must be able to be activated and deactivated while the visual system is running.

3.1.1.2 Shading

The system must provide Gouraud shading for rendered polygons.

Texture rendering is a computationally expensive activity that most smaller workstations cannot handle at acceptable frame rates. In order to support activities on these low-end systems, visual models will need to have sufficient polygons to be recognized without texture.
3.1.1.3 Visual Resolution

A minimal visual resolution of 1280 x 1020 pixels will be maintained in single-channel systems. Multiple channel systems will maintain a minimum resolution of 960 x 680 pixels in all channels.

These visual resolutions are the current industry standard for workstation display monitors. Some of the hardware vendors listed in section 4.0 have the ability to support other resolutions, but 1280 x 1024 is the norm for the image generation industry.

3.1.2 Visual Data

3.1.2.1 Visual Terrain

The following three levels of detail (LOD) must be supported for display of terrain.

-Ground based LOD
This LOD is for visual scenes where the visual eyepoint is less than 25m above the ground. This level will need to support a resolution suitable to infantry exercises as well as armored and support vehicles. This LOD should represent nearly perfect correlation with the databases used in the TESTBED CGF and ModSAF systems.

-Mid Level LOD
This LOD is typically used for low flying aircraft operating at less than 250m. Helicopters in cruise mode, Ground support fixed wing aircraft, and roving eyepoint mode will use this level to help reduce redraw jitter when rapidly traversing the database.

-High altitude LOD.
This represents the lowest level of correlation with the ideal terrain model. Typically, this LOD is used when the eyepoint is too high to distinguish ground targets or small cultural features. It will allow stable visual rates when the visibility range and field of view cover such a broad area that the polygon counts, for lower altitude LODs, would overload the image generator.

3.1.2.2 Visual Models

All visual models must contain a sufficient number of polygons so as to allow identification of vehicle type without the use of texture in the visual system.
Models used must support the generic set of appearance parameters as specified in section (3.3) of the DIS Enumeration Document [IST93a].

Elements of these visual effects shall be recognizable without the use of texture in the visual system.

The untextured recognition requirements are driven by the anticipated need to use the models on hardware that is not capable of hardware texture support. Software based texture is extremely expensive computationally, thus it is not normally used when a smooth frame rate is desired on these low-end systems.

3.1.2.3 Environmental Effects

No requirements are slated for system ability to render environmental effects.

Although many vendors do provide some capability, there is no sense of uniformity as to how these components will work. Due to immaturity of DIS requirements for visual environment, environmental effects have not been incorporated as part of the TESTBED Stealth requirements (see section 4 of the Test Procedures Document [Vanzant-Hodge94]).

3.1.3 Update Rate

A minimum visual frame rate of 15Hz is required.

This will provide a sufficient update rate to allow for the TESTBED and engineering simulation requirements. Lower frame rates, while sufficient for basic testing, do not demonstrate stability problems in entity movements. Such problems will often manifest in the form of a position/orientation jitter that would not be readily visible at a lower frame rate.

3.1.4 Image Capacity

A minimum rendering rate of 75,000 polygons per second is required.

For the purposes of this report, image capacity will be discussed in terms of polygons per second. This specification term is common throughout the industry and is the most readily used benchmark currently available. Based on the polygonal counts demonstrated in currently available visual databases, a capacity of 75,000 polygons per second has been found to suffice for DIS testing and simulation prototyping purposes. This figure is based on a 15Hz frame rate, utilizing databases from previous I/ITSEC DIS demonstrations that consist of textured polygons.
3.2 Interface Requirements

There are three interface components utilized in the TESTBED Stealth system. These interfaces include a DIS Interface which receives and processes the network traffic, a Visual Interface Manager which connects the DIS environment to the visual application, and a User Interface that provides for control over the visual scene and eyepoint placement. Each of these interfaces present certain requirements.

3.2.1 User Interface

The user interface on the Stealth system must allow for the following items:
- User Controlled Eyepoint Location
- User Controlled Eyepoint Orientation
- Entity Selection for Attachment
- Controls for Attachment Mode
- Environmental Settings
- Visual Scene Controls

This user interface must be reconfigurable within the development environment to support different experimental and testing efforts.

The interface will either operate as a separate application which interacts with a shared memory area containing visual control parameters or be an integral part of the visual application.

3.2.2 DIS Interface

The DIS interface must support the reception and maintenance of DIS entity and event information in a manner such that a separate visual application can observe and utilize the visually apparent aspects of the simulated environment.

Due to the continuing research into PDUs and DIS implementations, the interface must allow for rapid alterations and additions to handling DIS information.

The selection of a DIS interface package was driven in part by the hardware platform selection.
3.2.2.1 Network Capacity

The network interface must be able to handle a DIS packet environment with a mean load of 1000 packets per second. This is the level of traffic observed during the network load testing experiments [Cheung94] at the 1993 I/ITSEC conference. Under currently provided Unix System V implementations, an effective packet load limit of 500 packets per second has been observed.

It is, therefore, a requirement, that source code for the DIS Interface and platform network drivers be available to TRIDIS engineers.

3.2.2.2 Entity Capacity

The DIS interface module must be capable of processing and maintaining the DR modeling for no less than 250 entities with a desired target of 500 entities.

These numbers were driven by anticipated I/ITSEC '94 simulation loads and are in no way intended to reflect on the visual model capacity of the Stealth system. When entity counts begin to reach the 250 entity level, filtering schemes will have to be employed to manage the entity tables and only pass the most relevant entities to the visual system.

3.2.3 Interface Hardware

Any selected system must provide for optional support of the following types of system and network hardware. These components are anticipated to be requirements for other aspects of the TRIDIS project involving alternative network media protocols and configurations.

3.2.3.1 VME

The selected hardware must have either an actual VME bus slot or support a bus architecture which is supported by a COTS interface that will interconnect with a VME chassis.

This requirement is driven by the eventual need to utilize different types of hardware interfaces and other components of the TRIDIS project. There are several vendors currently supporting such devices with all classes of SGI hardware supported in one way or another.
3.2.3.2 EtherNet

The adopted system must be capable of driving at least two EtherNet ports, with the ability to handle three preferred.

This requirement is based on an entity environment where a market standard port will not be capable of receiving and processing the desired volume of traffic, or where the host system must be able to access Unix TCP/IP services such as NFS or NIS in order to function properly. This requirement is not expected to pose a problem with any hardware platform that is Stealth capable. Any planned usage of a secondary EtherNet port should be preceded by a careful analysis of the system impact and capabilities of different port configurations.

3.2.3.3 FDDI

The selected system must have the ability to be expanded to accommodate a Fiber Data Distributed Interface FDDI network device.

Optical EtherNet / FDDI interface options are currently under research both within IST and other organizations thus mandating that any system selected should be capable of supporting the installation and usage of a FDDI interface using the EtherNet protocol family.

3.2.4 Visual Interface Manager

The selected or authored system must support an effective system interface with a resident simulation’s runtime in terms of environment both hardware and software.

This interface must allow for sufficient information transfer to permit the system to act as the run time visual system for a manned training platform.

It must allow for the transfer of all target and visual queue information at a rate required by the runtime simulation environment.

Based on current industry capabilities, and in order to fit within the cost constraints of this project, the update rate will be assumed to have a lower bound of 15Hz and an upper bound of 60Hz. On the low end, this will support the required visual update rate of 15Hz (see section 3.1.3) required for the TRIDIS Stealth. For TRIDIS efforts that involve the construction of a manned reconfigurable simulator, the 60Hz rate will be sufficient for any planned aircraft simulation.
4.0 Product Survey

After the basic requirement set was distilled for this task, a survey of available products, both government and COTS, was undertaken with the following results. Due to time and facility restrictions, these surveys did not include in-depth examinations of each product. In most cases, the review included verification of claimed features that directly impacted DIS testing utility. The listed products represent what was commonly available when this task began (summer 1994), and does not include products introduced into the market since that time.

4.1 Hardware Platforms

During the review of potential hardware vendors, several hardware platforms were reviewed. These included Silicon Graphics, Kubota, Evans & Sutherland, and Intel based systems. The hardware evaluation process was not very in-depth due to the quick disqualification of most of the options. The Evans & Sutherland option was immediately disqualified due to cost and Kubota left the market in late summer of 1994. The PC based systems were and are still under investigation and show considerable promise for the near future. This left the Silicon Graphics platform as the most viable option. In addition, simulation management and DIS radio software under review by the project called for a SGI capability to be available. This greatly enhances the potential utility of an SGI selection. As most of the visual research activities at IST utilize Silicon Graphics platforms, this system would also allow the TRIDIS project access to existing software and expertise within IST.

4.2 Software Products

Several potential products were reviewed for each of the principal software components. The depth of these evaluations was driven by the availability of hardware, software license restrictions, and vendor support. Due to the relatively small number of candidates in each category and the needs of the TRIDIS effort, the reviews did not require an in-depth analysis of each candidate.

4.2.1 Visual Systems

The only visual software products reviewed were those implemented on Silicon Graphics. This was due to the fact that the high end image generators, such as Evans & Sutherland, are packaged solutions that do not allow for user alteration of the visual software. Low end visual scene generators which are gradually coming onto the market are based on the OpenGL library, discussed in section 4.2.1.2, and have been ported to Pentium processors from Silicon Graphics systems. When reviewing SGI based visual options there were only two categories of visual software packages, OpenGL and Performer.
4.2.1.1 Silicon Graphics Performer

Silicon Graphics provides for the rapid development of visual display systems through the use of an application programmers interface (API) called Performer. The Performer package is a simplified graphics library that supports the rapid generation of 3D applications such as a Stealth. The OpenGL package underlies most of the Performer development kit as the actual graphical programming language. This package is currently in use by a number of vendors and government contractors for both simulator visual systems and stealth displays.

4.2.1.1.1 Coryphaeus EasyScene

EasyScene is a user API which is written on top of the Silicon Graphics Performer utility. The combination of EasyScene and Performer allows the user to construct a visual system without having to write graphics code which can become quite complicated in many circumstances. Licenses were provided by Coryphaeus Software, free of charge, for a period of two years (anticipated expiration is January 1996). These licenses are for use at IST only and must be renewed monthly.

4.2.1.1.2 Paradigm

Paradigm Simulation produces a Performer based visual system that was briefly examined but not evaluated during the course of the Stealth task due to problems with license terms imposed by the vendor.

4.2.1.1.3 Mak Technologies VR - Stealth

The Mak VR-Stealth product was reviewed during the summer of 1994 when the Stealth task was initiated. At the time, this was the only product with any significant market history. Examination of VR-Stealth’s capabilities and support, along with the requirement that it be used with the VR-Link DIS interface product, made it unsuitable for the project. Articulation and Visual Effects required significantly more code to be written for support and the vendor had yet to release a functional scheme to handle these elements. Another problem with this product was a dependency on the MultiGen product which was not readily available to the project due to disclosure restrictions on donated licensees and a lack of funds to purchase a license set for the project. Many of the problems have been redressed since the initial evaluation and contact with Mak personnel has been maintained.
4.2.1.2 GL/Open GL

GL and OpenGL are visual development libraries supported by Silicon Graphics. These libraries represent the level of software architecture below the Performer product. There are currently several companies with GL/OpenGL based visual systems which outperform Performer based applications. These products were not selected for use in the TRIDIS project due to restrictions on time and personnel capable of making modifications required for use in the TESTBED. Currently, these products are based at a very high level of sophistication, and most are not readily modified from delivered configuration. As OpenGL becomes more of a market standard, tools (i.e. SGI Performer) to reduce the development time of OpenGL based applications are becoming available.

Some visual development companies have begun to port their visual applications to Intel Pentium based hardware using newly marketed graphics cards. Although these packages show great promise in performance and cost improvements, they are only now becoming mature enough to make their evaluation worth while. All of the visual development companies which were investigated under this project are listed in the following sections.

4.2.1.2.1 Gemini Technology

Gemini Technology produces a package called GVS which is based on the OpenGL graphics language. This software is currently running on Silicon Graphics hardware, but is expected to be released for the Pentium based computer market during the second quarter of 1995. IST personnel have expressed an interest in beta testing this package when available in an effort to determine if it is suitable for use as a low-cost stealth system. As yet, such an activity has no funding source.

4.2.1.2.2 Computer Explorations Inc.

Computer Explorations Inc. (CEI) also provides an optimized OpenGL based visual system. CEI is also in the process of releasing a Pentium based visual application on specially configured PC systems marketed by Intergraph. The vendor claims a 50,000+ textured polygon per second capability for approximately $30,000. This claim is based on the use of a dual Pentium processor system and has not been validated by IST.

4.2.1.3 Loral 2D/3D Viewer

During the early stages of the Stealth task, a copy of the 2D/3D Viewer developed by Loral ADS was provided by the TRIDIS COTR for evaluation. The architecture of the package was not reviewed, but after consultations with AccuSoft personnel, the package was run on
the Onyx system obtained by the TRIDIS project. Due to the limited configuration of this Onyx, the application proved unsuitable for use. This appears to be caused by the requirement that both the 2D and 3D applications operate when a Stealth view is desired as well as by an architecture that was not optimized for the SGI. Discussions with the authors and examination of the code indicate it was designed to be cross platform compatible with less visually capable hardware, leaving many SGI optimization elements un-utilized. Later discussion revealed that these problems would be addressed in the future, but for 1994 I/ITSEC development and testing a fully configured Onyx would be required to obtain suitable performance.

4.2.2 Visual Modelers

For the purpose of this report, Visual Modelers are software applications used to generate three dimensional icons for use in a visual display system (Stealth). These modelers take the form of specifically tailored variants of more traditional computer aided design (CAD) systems used for industrial 3D surface modeling. For the purpose of the TRIDIS project, three packages were examined from a utilitarian perspective with brief descriptions provided below. Due to time restrictions, a detailed trade study was not performed; however, within the sections of this report discussing selected products it will be shown why such a study would not have impacted the decisions made.

4.2.2.1 Designers Workbench - Coryphaeus

Designers Workbench is a visual modeling application that allows the rapid construction of 3D models complete with articulation and visual effects. The control for these components is provided within the EasyScene visual application and requires the user to provide code which copies the relevant information from the DIS interface to memory locations provided by the run time system.

4.2.2.2 MultiGen

Multigen is a 3D modeling package commonly used in industry to produce both terrain and model databases. The package is a standalone product which provides polygonal model files in the Flight data format, which is currently the most commonly used format in the simulation market. This package was not used in the production of any materials generated by the TRIDIS project due to license restrictions that would have placed IST at conflict with non-disclosure restrictions required by sponsors that provided visual models. It was, however, used to inspect databases provided by outside sources for use at the I/ITSEC conference.
4.2.2.3 S1000 - Loral/BBN

The S1000 modeling package was developed by BBN systems and is currently under the control of Loral. Although still operational within IST, the package has not been significantly updated in a number of years. While running under version 4.0.5 of the Silicon Graphics operating system (Irix), the application suffered repeated problems due to memory corruptions likely due to incompatibility with changes in the operating system. While this package is still used by IST to generate databases for use in IST held SIMNET resources, it is not readily compatible with visual packages under review.

4.2.3 Terrain Modelers

Terrain modeling tools are used to build the surface skin which constitutes the terrain database. Most of the products currently in use within industry will translate Defense Mapping Agency (DMA) and U.S. Geological Survey (USGS) data sets into a polygonal format that can then be processed with surface features such as trees, roads, and structures. Although there are a number of tools within industry to handle this type of work, only two were found to be on the market and focused to the visual simulation industry. These two products are described below.

4.2.3.1 EasyT - Coryphaeus

This is a terrain editor/generator that can read and process DMA and USGS data sets as well as create terrain databases from scratch. The data structures and utilization information for this tool support textured terrain skins in multiple formats with instantiated surface features. Two licenses were made available to IST for an indefinite period of time. A full evaluation of the tools capabilities has yet to be performed due to a lack of funding for terrain database creation.

4.2.3.2 MultiGen

MultiGen terrain editing is currently under development and, as of summer 1994, lacked many features needed for rapid database development. Many of these problems are under repair but cost remains a major problem with this package. The Visual Systems Laboratory of IST currently utilizes this software, but restrictions placed on the licenses by MultiGen preclude the use of those systems for TRIDIS purposes other than simple database review.
4.2.4 DIS Interfaces

In section 3.2.2.3 of the TRIDIS workplan (IST-TR-94-17) the Mak VR-Link DIS interface package was specified for use in the TRIDIS Stealth effort. This initial plan was modified due to a need for the rapid inclusion of experimental PDU research and testing without the dependency of Mak updates. Several Government funded packages were reviewed on an unsupported, no-cost basis. These packages, along with the Mak product are discussed below.

4.2.4.1 DIP (NAWC/TSD Aladdin)

The DIS Interface Package (DIP) was developed by NAWC/TSD as part of a visual package called Aladdin. This package is still in use and IST is in possession of the source code. This source code has limited flexibility as it was written for a specific application but has proved useful in experimental efforts both before and during the I/ITSEC 1994 conference when Stealth PDU work was taking place. Modified by Gary Hall, currently with Coryphaeus Software and the author of Aladdin, it was adapted to use the Stealth PDU (described in section 6.2.2.1 below) in an effort to test the IST Stealth package and for the use of Navy personnel.

4.2.4.2 Mak Technologies - VR-Link

VR-Link, by Mak Technologies, is currently the commercial standard for DIS interfaces marketed to industry. This interface is intended for rapid user integration with an existing simulation application without requiring extensive knowledge of DIS on the part of the developer. While an API package is available from Mak which allows user written overlays to be compiled into the VR-Link package, source code is not normally provided and was not obtained by the TRIDIS project. Contact with Mak personnel has been maintained in an effort to coordinate DIS interface issues including the Stealth PDU and visual interfacing. This package is in use by other projects within IST that do not have a need for experimentation with the DIS protocol.

4.2.4.3 NRaD SGI Application Interface

Naval Research and Development (NRaD) personnel have developed and maintained a set of DIS interface tools that are currently in use by a number of government research and defense contracting entities. These tools were originally developed to operate in the VxWorks development environment using VME target board systems. Subsequent development allowed the code to be executed in both the Sun and SGI development environments. In the case of the SGI, the DIS communications task was running either within the SGI environment or on an installed VME processor board located within the SGI system. This interface has been
upgraded to DIS v2.0.3, and all source code and libraries have been provided to IST on an unsupported, unrestricted basis by Kevin Boner of NRaD.

4.2.4.4 Loral Interface Library Developers Kit (ILD)

The ILD library, developed by Loral ADS, was not reviewed as part of this effort. Based on the literature available, the library is overly complicated for the purpose of the Stealth task. It appears to be focused on the task of protocol translation and cell adaptation rather than simple DIS reception and entity tracking. This package would be worth further investigation should both SIMNET & DIS requirements be imposed on future efforts.

5.0 Final Selections

When making the selections for system components, several additional factors were considered.

- The system needs to represent maximum flexibility in capabilities;
- Minimal 'graphics' coding should be required by TRIDIS personnel;
- The system should be able to visually represent all required DIS 2.0.3 visual parameters [IST93a] without the need for explicitly coded routines for each model and visual effect;
- A reconfigurable user interface should be part of the final configuration;

5.1 Visual System

Based on the criteria listed above, as well as on cost and availability issues, the decision was made to proceed with a SGI Performer based visual system, using a commercial application integrated into the selected DIS interface. The application selected was the EasyScene product which was provided free of charge by Coryphaeus Software for use within IST. This selection was based primarily on the ease of integration and the low code development times required for the integration of DIS visual parameters into the visual scene. Due to the use of the SGI Performer base, databases from most recognized vendors are compatible and direct knowledge of the internals of the SGI visual libraries was not required for TRIDIS personnel. In addition to these other criteria, the EasyScene product supports the use of Designers Workbench, Dynamic Run Time (DRT) modules which contain a visual model, along with all of the animate visual effects in a format that can be easily updated during run time.
5.2 Visual Modeler

The selection of a visual modeling package followed the same criteria as the visual display software. It was desired that DIS developmental personnel not be required to have a proficiency with visual software development while still being able to modify the appearance processing of visual models within the Stealth system. This fact, along with the no cost availability of the software to the TRIDIS project, drove the selection of the Designers Workbench package by Coryphaeus Software. This package allows the developer to edit a visual model and incorporate appearance changes and articulation control via a shared memory architecture, that, when loaded into the EasyScene product discussed in section 4.2.1.1.1, will support interface access for updates. This model format will also be supported in the Gemini OpenGL product in June 1995, allowing the use of the same visual models for investigations into PC based stealth display devices.

5.2.1 Visual Models

The time required to prepare visual models or icons can be considerable depending on the resolution of the models. Based on schedule and cost constraints and the poor quality and availability of models in the public domain, the TRIDIS project approached outside commercial and government sources in search of a model library suitable for ‘show quality’ Stealth display purposes.

Viewpoint Datalabs has provided IST access, on a non-disclosure basis, to their library of military models. These models provided the starting point for the TRIDIS modeling efforts. Source models, which contained texture and geometry were modified to provide articulation and visual effects. The models were tested prior to, and demonstrated successfully at, the I/ITSEC 1994 conference.

5.3 Terrain Modeler

Terrain modeling was beyond the scope of the TRIDIS Stealth effort and a search for suitable terrain databases provided only limited results. As part of the tool set donated to IST by Coryphaeus Software two EasyT licenses were provided to the TRIDIS project. This tool has not been extensively utilized or evaluated by IST but it appears to meet all of the previously stated requirements. During the course of I/ITSEC preparations SGI offered to provide a visual database for show participants thus removing the immediate need for IST to generate one internally. The ability to generate terrain databases will need to be present to support DIS testing in the future. A generic test terrain database including terrain patches specifically generated to support testing with specifically generated geometry that supports known parameters will allow for more closely controlled, accurate DIS compliance and interoperability testing.
5.4 DIS Interface

The NRaD AIU package was selected for use in the TESTBED Stealth system due to source code availability and architectural stability. The package is run using a minimal configuration with capacities and update rates configured prior to compilation. The other DIS interfaces listed in the review section were not selected due to a lack of source code availability (Mak) or because they provided for less suitable architectural features. While the NRaD-AIU package is currently under use, the current use of a Unix network port means that the 500 packet per second limit is in force. It is recommended that further investigation be made with regard to a faster port and driver (see section 8.3).

5.5 User Interface

User interfaces were not extensively evaluated due to the high cost associated with most graphical user interface (GUI) development tools. Based on the GUI interface used in the Aladdin Stealth package, the Forms package was selected. This package supports graphical construction of control interfaces and will generate C code files for compilation into the target application. The GL version of this package is in the public domain for non commercial purposes and was obtained from the Utrecht University (Netherlands) anonymous FTP distribution node (cs.ruu.nl). The base GUI control panel used in the TRIDIS Stealth effort was taken directly from Aladdin with some modifications made by Gary Hall of Coryphaeus systems. These modifications consisted of the replacement of outdated GL library calls with EasyScene command scripts.

6.0 Development Issues

6.1 Visual System

6.1.1 Visual Model Loading

In addition to the database load times, the entity and structure models also presented problems with load and processing times. In the current implementation, when the first instance of an entity is detected by the Stealth, the data for the visual model for that entity type is loaded into the system and the run time processing module is built. This process typically requires five to eight seconds. During many of the demonstrations at the I/ITSEC conference, typical model counts ranged from 75 to 120 active entities. The time required to load that many models is considerable. In addition to the problem presented by models, there is the problem of the time it takes to handle visual effects and weapons fire. By the time this activity is added up, the Stealth system faces considerable loading problems without taking into account
the terrain database. The repeated loading and unloading of visual models also triggered one of the current problems with the Silicon Graphics Performer library. This problem is a memory allocation error that, under a heavy load, can cause the allocation heap to fail. This problem occurred several times during the I/ITSEC show in several different booths. Conversations with the Silicon Graphics engineers revealed that they are aware of the problem which is based in the current version of the operating system (Irix 5.2). This version of Irix should be replaced in the second quarter of 1995. Software vendors are working on patches to fix the problem. TRIDIS personnel intend to wait for SGI to fix the problem. The TRIDIS Onyx, although suffering several crashes during the show, does not have enough RAM to load the number of visual models required to begin encountering the problem frequently.

6.1.2 Terrain Loading

Several problems have arisen from system loading issues. These problems are centered around visual database size and structure along with the inefficiency of the current file loading systems. Prior to the I/ITSEC demonstrations, Coryphaeus provided a utility that allows visual files to be preprocessed into an internal binary format that can be ‘fast loaded’ into the Silicon Graphics system. This utility takes a memory image resident in the SGI Performer system and dumps it into an optimized file. In order to use the facility, a Performer loader for the source format must be used to get the file into memory.

When IST arrived at the conference, a new database was provided by Silicon Graphics and MultiGen which contained a minor format change that prevented proper preprocessing by the Coryphaeus utility, forcing IST personnel to use the file loader provided by MultiGen. This loader requires three to five minutes to load the visual database rather than the minute or less the Coryphaeus fast file loader required. This meant that every time the system needed to be restarted, five to seven minutes was required. The problem with the loader is being handled by Coryphaeus, and a revised version should be ready by the time this report is delivered. The issue of database load times is also being addressed by Silicon Graphics and a solution is scheduled to be released in June 1995 with the 2.0 version of the Performer product.

6.1.3 Terrain Databases

The principal loss of system performance came from poorly correlated terrain databases that were available for use on the Silicon Graphics machines. Due to time and money restrictions, IST was forced to utilize terrain databases that were being provided to I/ITSEC conference attendees free of charge. This restricted the project to four database choices.
The 'Bwanavision', Hunter-Liggett Database
This database was generated by Bob Buckely from SIF terrain data files for use in the Visual Systems Lab in IST.

1993 I/ITSEC Marconni Database
This database has numerous errors in terrain feature accuracy and continuity. It has been placed in the public domain as part of the Silicon Graphics Performer v1.2 demonstration/resource kit.

1994 I/ITSEC Marconni Database
This is a revised version of the 1993 effort that was commissioned by Silicon Graphics. Several months before the 1994 conference, however, SGI abandon this effort in favor of a completely new database.

Paradigm - Multigen Hunter-Liggett Database
This database was commissioned as a joint effort between Silicon Graphics, Paradigm, and Multigen. The exact control and funding structure of this effort has not been made available to IST. Approximately four to six weeks prior to the 1994 I/ITSEC conference, this effort was discontinued prior to completion for reasons not known to IST.

After the Paradigm-Multigen database was discontinued, Silicon Graphics and Multigen personnel resumed work on the 1994 version of the database that Marconni had been working on. This effort focused on removing errors in the terrain as well as adding cultural features required by demonstration coordinators. Silicon Graphics then proceeded to distribute this database to conference attendees with the requirement that a MultiGen notice be posted next to any display device utilizing the database. This database was the only one of the choices that provided the correct cultural features for visualizing the principal demonstration areas. It had a mean elevation error of -0.9 meters [Nelson95] which forced the use of terrain clamping (see section 6.1.4) for ground entities. In deference to the provider, the database was intended for air operations, not ground forces. Of the other three databases that were available, one had a similar error level and did not contain the airport which was the focus site for most of the demonstrations. The other two, which had significantly better correlation, had even fewer features referenced for visualization. These problems were the result of dueling database generation criteria. Almost all of the databases originated from the S1000 Hunter-Liggett data that had been processed into the SIF format. The companies with high-end visual systems (i.e., E & S and BBN) and those not operating a 3D visual display, processed databases with either a high resolution sampling distance or a direct polygonal conversion, while other companies utilized tools within the SIF toolkit to resample the terrain at a significantly lower resolution, with the intention of gaining faster frame rates. This methodology is fine for exercises that are restricted to air operations with no moving ground targets, but is quite flawed if ground operations are to be employed due to the loss of terrain resolution.
6.1.4 Ground Clamping

Due to problems with terrain correlation in available databases, single point ground clamping was employed. This type of ground clamping acts only upon the elevation of an entity and not upon its orientation. With terrain clamping in place, the visual update rate fell to 10Hz while it remained stable at 25Hz without clamping. This performance drop was observed without regard for the number of entities present. Personnel from the SGI Performer engineering group stated that this is due to the method used to clamp models. SGI plans to release a different scheme for clamping in the 2.0 release of Performer but, without a paging/partitioning scheme in the database, ground clamping will continue to be an inefficient tradeoff with respect to polygon count. Future interoperability efforts need to include these considerations in the planning phase and set much more detailed terrain data requirements. Figure 1 illustrates this form of clamping.

![Single Point Ground Clamping](image)

**Single Point Ground Clamping**

Single point ground clamping affects the elevation component of an entity's local coordinate. This compensates for miscorrelations in the elevation of the polygons that constitute the terrain surface.

*Note:* The orientation of the entity is not corrected in this mode.

6.2 DIS Interfaces
6.2.1 Stealth PDU Experiments

As part of the experimental aspect of the 1994 I/ITSEC demonstration, an effort was made to provide for master and slave stealth systems. Through this mechanism, a single stealth operator would fly the visualization point while the other Stealth systems tracked along. After several discussions as to how this should be accomplished [Humphrey94], it was decided that the data structure for an Entity State PDU would be used as a Stealth PDU with modified field values. These values are described in the following sections.

6.2.2 Stealth PDU Utilization for I/ITSEC '94

To allow for coordinated demonstrations with multiple visual stealth systems and without requiring large numbers of technical personnel to manually operate the equipment, an experimental Stealth PDU was utilized for the I/ITSEC '94 demonstration efforts. This PDU was intended to provide the necessary functionality by permitting a stealth system to mimic a remotely operated stealth entity, operated by demo coordinators during key points in the exercises.

6.2.2.1 Experimental Stealth PDU

The following sub sections detail the enumeration requirements for the experimental Stealth PDU as set out for participants of the 1994 I/ITSEC experiments.

6.2.2.1.1 PDU Header

Header fields operate in the same fashion as those in the Entity State PDU with the exception of the PDU type which will be 130 decimal.

6.2.2.1.2 Force ID

The force ID is set to Neutral (3 decimal).

6.2.2.1.3 Entity Type

The Entity type fields are not utilized and should all be set to zero.
6.2.2.1.4 Entity Position and Velocity

Entity location and velocity are processed in the same manner as with any of the vehicles. Dynamics models should be maintained to allow for meaningful velocity vectors for use by the dead reckoning processors. However, orientation dependencies that may be present in individual DR implementations should be re-evaluated for the Stealth PDU information as the field will not necessarily relate to direction of travel.

6.2.2.1.5 Entity Orientation

Orientation should be utilized for the purposes of look angle only. The vector provided indicates the direction in which the eyepoint should be centered. In order to support a smooth visual pan, the angular velocity DR will still be required, but should be processed independently from the remainder of the velocity and acceleration information.

6.2.2.2 I/ITSEC Experimental Stealth PDU Problems and Recommendations

During the I/ITSEC demonstration, the implementation of the experimental Stealth PDU, described above, was attempted. The PDU was populated with position and orientation information along with basic dead reckoning parameters. The object was the 'slaving' of one or more stealths to a master eyepoint with reasonable visual stability. This was basically successful, but problems with the design of the slaving mechanism caused the visual scene to be highly unstable in many circumstances.

6.2.2.2.1 Latency Mismatch Jitter

When the master visual eyepoint is attached to a particular entity, a slave eyepoint, which is looking at the same mathematical location as the master, is running with location information that is several frames older than the DIS environment the master is operating in. For example, the master eyepoint locks to a wingman's view of an F16 aircraft moving at MACH one and begins to transmit the location and orientation of the eyepoint. The slave stealth is locating itself at an eyepoint coordinate that is up to 100ms old and is based on an entity whose position when looked at by the master is up to, and maybe over, 100ms old. Thus, it is looking at the position of the aircraft that may be over 200ms old. In 100ms, the aircraft, has moved more than 60 meters away from the location that the master stealth broadcast. As the dead reckoning and update rates are not normally equidistant on the timeline, a jitter in the visual scene occurs (see Figure 2). The degree to which this occurs depends on the velocity of elements within the scene. This problem becomes more pronounced if multiple systems are acting as master and attempt to connect to other stealths in a chain.
With A and B in the visual system, the moving eyepoint is processed via a dynamics module, into an outbound dead reckoning routine and converted into DIS coordinates. The network stealth eyepoint now contains three coordinate conversions and three dead reckoning stages.

The network stealth eyepoint now contains three coordinate conversions and three dead reckoning stages. Vehide from Source A

Vehicle from Source B

Figure 2. Latency Mismatch Jitter

6.2.2.2.2 Offset Stacking

Another problem with the I/ITSEC 1994 version of stealth slaving is the cumulative error incurred when positional information is passed through multiple coordinate conversions and dead reckoning packages. Each iteration has an associated error which is added to any previously acquired inaccuracy. This is illustrated in Figure 3.

Observer Attach or 'Tethered' Mode

The 'Slaved' Stealth is provided an entity attach ID as a base point and positional and rotational offset information via a Stealth Control PDU that are reflected in the eyepoint. The eyepoint is a product of the two sources of information. This mode has the effect of placing the eyepoint into the body coordinate system of the attached entity.

Figure 3. Example of Offset Stacking
6.2.3 The Stealth Control PDU

In an effort to resolve the problems with the I/ITSEC 1994 experimental Stealth PDU, a Stealth Control PDU (SCPDU) is currently under discussion. A preliminary description of this PDU is provided below. The premise for this PDU is that by telling slave stealth systems to attach to a particular entity, rather than the master attaching and then transmitting its position, the jitter problems would be greatly reduced.

The Stealth Control PDU shall communicate configuration and control information to designated Stealth visual systems. This information shall include modes of visual attachment and base point and eyepoint control as well as basic visual display configuration data.

6.2.3.1 Conceptual Model for Stealth Control

The placement and control of a stealth eyepoint will be based on two control points. These control points will be the Stealth Base (SB) and the Stealth Eyepoint (SE). The Stealth Base is used as the attachment point for the 'camera.' It can be operated either as a free flying 'carpet' or it can be attached to an entity within the DIS environment. In both situations, the SB is simply a point in space without any orientational parameters. Within the SCPDU, position and linear velocity are the only two parameters needed to designate the SB and when unattached, the SB is always dead reckoned using algorithm number two.

The SE has multiple modes of operation. These eyepoint modes are based on the attachment status of the SB and SE. The operational categories currently under consideration are listed below along with sample modes of operation.

Base Unattached, Eyepoint Unattached:
This is free fly mode. The SB location and velocity describe the eyepoint's basic movements, while the 'Entity Attached Orientation Offset' and 'Eyepoint Angular Velocity' fields are interpreted as they would be for a normal entity.

Base Unattached, Eyepoint Attached:
This is a 'Tracking' mode. The SB may be positioned freely using world coordinates, but the eyepoint is fixed such that the entity described in 'Eyepoint Attach Entity ID' is centered within the field of view. With this class of attachment modes, all of the eyepoint position descriptors are relative to the entity to which the eyepoint is attached.

Base Attached, Eyepoint Unattached:
There are two modes that fall into this category. The first is a 'Tethered' mode. The SB is attached to a particular entity with the SB locator fields ignored so long as the 'Base Attach
Entity ID’ described entity is visible on the net. Should this entity not be present, these fields will be referred to for SB guidance. While is this mode, the eyepoint may roam freely.

The second mode of operation is the ‘Cockpit’ or ‘Mimic’ mode. In this mode, the SB is attached to a particular entity with a body coordinate that moves the eyepoint into a position representative of a crew members’ or instruments’ perspective (see Figure 4). This perspective is maintained as it would be for the actual crew member with orientational changes mimicking the movements of the entity.

**Cockpit or ‘Mimic’ Mode**

The ‘Slaved’ Stealth is provided an attach ID and offset information via a Stealth Control PDU. The eyepoint is a product of the two sources of information. For this mode of operation, either the issuer of the Stealth Control PDU or the Stealth itself will need information as to where the eyepoint is to be placed so that is does not conflict with the actual visual model display in the stealth. (i.e. if the eyepoint is within the model, the display may become obscured by backfaced polygons.)

*Figure 4. Mimic Attachment Mode*

**Base Attached, Eyepoint Attached:**

This situation is the combination of the two proceeding mode descriptions. There are two ways that this mode is utilized. The first mode is a ‘Compass’ or ‘Orbit’ mode. Described in Figure 5, this mode places the eyepoint into the body coordinate system of the attached entity and focuses the visual scene on that entity. The resulting effect allows the user to freely fly around in the visual scene while maintaining a visual fix on a particular entity.

Another implementation of this type of eyepoint control is the ‘Pilot-Track’ or ‘Wingman’ mode. In these modes, the eyepoint is placed in body coordinates of the attached entity, but the visual scene is focused on a different entity. This provides the perspective of an observer mounted on one entity visually tracking another (see Figure 6). A real world example of this mode would be the televideo display that a laser designation pad provides on a bombing run. The pod will remain fixated on the target while the perspective is attached to the tracking aircraft.
'Compass' or 'Orbit' Mode

The 'Slaved' Stealth is provided an entity attach ID and positional offset information via a Stealth Control PDU. The orientation of the eyepoint is continuously focused on the same entity. Positional offsets are provided in entity body coordinates. The implementation of this mode is the same as that of the 'Pilot - Track' with the exception that the same entity ID is provided in the 'Entity ID for Eyepoint Attach' field in the Stealth Control PDU.

![Diagram of Orbit Attachment Mode]

Location of Entity
(or Stealth Base Point)
in World Coordinates

Location of Eyepoint in Body Coordinates
Orientation is controlled so that the attached entity is always center screen.

Figure 5. Orbit Attachment Mode

'Pilot - Track' or 'Wingman' Mode

The 'Slaved' Stealth is provided an entity attach ID and positional offset information via a Stealth Control PDU. The orientation of the eyepoint is continuously focused on the entity identified in the 'Entity ID for Eyepoint Attach' field in the Stealth Control PDU. Positional offsets are provided in entity body coordinates.

![Diagram of Wingman Attachment Mode]

Location of Entity
(or Stealth Base Point)
in World Coordinates

Location of Eyepoint in Body Coordinates
Orientation is controlled so that the eyepoint attached entity is always center screen.

Figure 6. Wingman Attachment Mode
<table>
<thead>
<tr>
<th>Field</th>
<th>Stealth Control PDU Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>96</td>
<td>PDU Header</td>
</tr>
<tr>
<td></td>
<td>Protocol Version__8-bit enumeration</td>
</tr>
<tr>
<td></td>
<td>Exercise ID__8-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>PDU Type__8-bit enumeration</td>
</tr>
<tr>
<td></td>
<td>Protocol Family__8-bit enumeration</td>
</tr>
<tr>
<td></td>
<td>Time Stamp__32-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Length__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Padding__16 bits unused</td>
</tr>
<tr>
<td>48</td>
<td>PDU Originator ID</td>
</tr>
<tr>
<td></td>
<td>Site__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Application__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Entity__16-bit unsigned integer</td>
</tr>
<tr>
<td>48</td>
<td>PDU Destination ID</td>
</tr>
<tr>
<td></td>
<td>Site__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Application__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Entity__16-bit unsigned integer</td>
</tr>
<tr>
<td>192</td>
<td>Stealth Base Location</td>
</tr>
<tr>
<td></td>
<td>X-Component__64-bit floating point</td>
</tr>
<tr>
<td></td>
<td>Y-Component__64-bit floating point</td>
</tr>
<tr>
<td></td>
<td>Z-Component__64-bit floating point</td>
</tr>
<tr>
<td>96</td>
<td>Stealth Base Linear Velocity</td>
</tr>
<tr>
<td></td>
<td>X-Component__32-bit floating point</td>
</tr>
<tr>
<td></td>
<td>Y-Component__32-bit floating point</td>
</tr>
<tr>
<td></td>
<td>Z-Component__32-bit floating point</td>
</tr>
<tr>
<td>32</td>
<td>Visual Range</td>
</tr>
<tr>
<td></td>
<td>32-bit floating point</td>
</tr>
<tr>
<td>32</td>
<td>Field of View</td>
</tr>
<tr>
<td></td>
<td>32-bit floating point</td>
</tr>
<tr>
<td>32</td>
<td>Time of Day</td>
</tr>
<tr>
<td></td>
<td>32-bit floating point</td>
</tr>
<tr>
<td>32</td>
<td>Haze Factor</td>
</tr>
<tr>
<td></td>
<td>32-bit floating point</td>
</tr>
<tr>
<td>16</td>
<td>Attach Mode</td>
</tr>
<tr>
<td></td>
<td>16-bit record of enumerations</td>
</tr>
<tr>
<td>48</td>
<td>Entity ID for Base Attach</td>
</tr>
<tr>
<td></td>
<td>Site__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Application__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Entity__16-bit unsigned integer</td>
</tr>
<tr>
<td>48</td>
<td>Entity ID for Eyepoint Attach</td>
</tr>
<tr>
<td></td>
<td>Site__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Application__16-bit unsigned integer</td>
</tr>
<tr>
<td></td>
<td>Entity__16-bit unsigned integer</td>
</tr>
<tr>
<td>48</td>
<td>Entity Attached Eyepoint Offset</td>
</tr>
<tr>
<td></td>
<td>X-Component__32-bit floating point</td>
</tr>
<tr>
<td></td>
<td>Y-Component__32-bit floating point</td>
</tr>
<tr>
<td></td>
<td>Z-Component__32-bit floating point</td>
</tr>
</tbody>
</table>
6.2.3.2 Information contained in the Stealth Control PDU

The Stealth Control PDU shall contain the following information.

a) **PDU Header** - This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header Record.
b) **PDU Originator** - Identification of the PDU's originator. This field shall be represented by an Entity Identifier Record.
c) **PDU Destination** - Identification of the PDU's target stealth. This field shall be represented by an Entity Identifier Record.
d) **Stealth Base Location** - This field shall specify the base location, for the stealth, in the simulated world. This field shall be represented by a World Coordinates Record.
e) **Stealth Base Linear Velocity** - This field shall specify an stealth base linear velocity. The coordinate system for an entities' linear velocity depends on the dead reckoning algorithm used.
f) **Visual Range** - The visual depth or display range desired in the visual display. This field shall be represented by a positive 32-bit floating point number.
g) **Visual Field of View** - The angular Field of View for the visual system. This field shall be represented by a positive 32-bit floating point number.
h) **Time of Day** - The Time of Day to be used in generation of ambient light levels. This field shall be represented by a positive 32-bit floating point number representing seconds.
i) **Haze Factor** - A numerical Haze Factor to be used in calculating the prevailing visibility. This field shall be represented by a 32-bit floating point number. (The generation of this factor is still under discussion)
j) **Visual Attachment Mode** - The mode of visual attachment to be used. This field is divided into two parts. The eight most significant bits describe the attachment mode for the Stealth Base with the remaining eight bits enumerating the attachment mode for the Eyepoint.
k) **Entity for Base Attach** - The entity identification to which base point is attached. This field shall be represented by an Entity Identifier Record.

| 96 | Entity Attached Eyepoint Orientation Offset | Psi 32-bit floating point
|    |                                           | Theta 32-bit floating point
|    |                                           | Phi 32-bit floating point
| 96 | Eyepoint Velocity                         | Entity Linear Velocity 3x32-bit floating point
| 96 | Eyepoint Angular Velocity                 | Entity Angular Velocity 3x32-bit floating point

Table 1 - Stealth Control PDU
1) **Entity for Eyepoint Attach** - The entity identification to which the eyepoint is focused. This field shall be represented by an Entity Identifier Record.

m) **Entity Attached Eyepoint Offset** - The positional offset between the base point and eyepoint. This field shall specify the location of the eyepoint in the attached entity's coordinate system. This field shall be represented by an Entity Coordinate Vector Record.

n) **Entity Attached Eyepoint Orientation Offset** - The orientational offset between the eyepoint and the attached entity. This field shall be represented by an Euler Angles Record.

o) **Eyepoint Velocity** - The velocity of the eyepoint in entity coordinates relative to the Eyepoint Attachment Entity. This field shall be represented by a Linear Velocity Vector Record.

p) **Eyepoint Angular Velocity** - The angular velocity of the eyepoint in entity coordinates, with regard to the Eyepoint Attachment Entity. This field shall be represented by an Angular Velocity Vector Record.

7.0 **Current Stealth Capabilities**

The following sections describe the current status of the TRIDIS Stealth system. Additional upgrades and enhancements are in progress, however they will not be completed in time for the required March 1995 delivery date.

7.1 **Visual Scene Control**

Visual scene control is accomplished via a GUI that was developed using the Forms GUI building utility. This panel is depicted in Appendix A along with a functional description of the controls provided. Shown in the top section of the panel is a list of currently active entities along with descriptions of each entity's type. Each entity is listed by site, host, and entity ID. To attach the Stealth eyepoint to a particular entity the user must select the entity from the list by clicking on it. Further information on supported attachment modes is also provided in Appendix A.

Positional control of the Stealth eyepoint is accomplished through the use of the 'Viewpoint Positioning' controls seen in the middle section of the GUI panel. These controls support full positional and orientational control of the eyepoint either in free fly mode or as an offset to the attached entity.

Other controls provide control over the field of view (FOV), the visibility range, and the time of day in the visual scene. These controls are all 'sliders' that are actuated through the use of the mouse. Visual system performance information and control over the video scan rate are provided at the bottom of the panel. Detailed descriptions of these functions are provided in Appendix A.
7.2 DIS Visual Appearance Support

All of the appearance bits described in section 3.3.1 (General) of the 2.0.3 Enumeration Document are supported in the run time software. Each visual model that is to be utilized in the Stealth must be edited to include the necessary texture and geometry required for each appearance alteration. Should additional appearance factors be desired, a code modification would be required in the visual interface package. The TRIDIS Stealth provides full support of DIS 2.0.3 appearance bits for the models for all of the entities for the I/ITSEC '94 DIS Interoperability Demonstration.

7.3 DIS Articulation Support

Visual model articulation is currently supported in the TRIDIS Stealth system. Each model that is to support articulation must be edited to include information specifying the location and plane of rotation that is to be used for each articulated component. At present, there is no defined limit as to the number of articulated components supported in each model as this is a system load issue. No code modification is required when adding additional articulation components. The TRIDIS project currently has all of the specified models from I/ITSEC '94 with the first order rotational parameters supported appropriately.

7.4 Stealth Remote Control

Stealth remote control is currently limited to the attachment of the local Stealth system to another unit that is broadcasting a ‘Stealth PDU’ under PDU kind 130. The TRIDIS Stealth is currently capable of broadcasting this PDU with a DR algorithm of 2. Additional refinements to Stealth control are discussed in section 6 of this report, but are not currently available for release.

7.5 Support for Audio System

Crystal River Sound System - The IST stealth system includes an interface to the AcoustitronII sound system. One of these systems was loaned to IST by Crystal River, for use at I/ITSEC '94 and has since been returned. The software interface is now being supported from within the EasyScene package, with the appropriate Crystal River hardware it can be re-installed on short notice.
8.0 Anticipated/Desired Enhancements to Stealth Package

During the course of this project, several enhancements have been identified as desirable for future testing efforts. These have been listed below, along with brief descriptions of the intended implementations.

8.1 Graphical User Interface for DIS Interface

To facilitate a greater ease of use in the DIS interface, the GUI controls currently used to control the visual scene should be expanded to control the DIS interface as well. Part of this work has already been started with GUI controls for exercise id and domain filtering already added to the panel. These controls have yet to be integrated into the interface.

8.2 Implementation of Remote Eyepoint Control

During the preparations for the I/ITSEC '94 conference, much of the work required to implement Stealth remote control was undertaken. The results of this work indicated the need for a Stealth Control PDU which is defined in a preliminary format in this report. This functionality needs to be implemented and refined so that it may be formally presented to the appropriate DIS working groups for review.

8.3 Migration of DIS Interface to High Speed Target Processor

One of the problems with the current configuration of the Stealth is a very limited throughput in the network port and DIS interface. DIS networking is currently accomplished through a generic Unix socket interface that bottlenecks at fewer than 500 packets per second on the TRIDIS Onyx. Furthermore, the overhead of managing this port and processing the DIS environment is overly burdensome to the level of hardware used in testing. A recommended fix to resolve this problem, would be to move the DIS interface to a VME target processor that would be located within the SGI chassis, and run independently from the visual system through a memory interface. Such a target processor would be less expensive than additional SGI processor capacity, and would have a much higher traffic capacity.

Currently, there are three DIS interface applications that could be used on a target processor. The first is the same DIS interface used to date on this project. It was originally designed for the VME environment and could be returned to a VME target processor configuration. The second interface is a joint Navy/Motorola package called the Network Interface Unit (NIU) that will be distributed through the Tactical Warfare Simulation and Technology Interface Analysis Center (TWSTIAC) at IST, in the summer of 1995. This product has been in use for a number of years in different forms by various government contractors and agencies.
The last of the three is a product produced by McDonnell Douglas Training Systems (St.Louis, MO) called the ‘DIS Engine’. Both of the Navy products have the ability to run either within the SGI or on a target board. This would allow a single VIM to connect with either option without major modification.

8.4 Improved Visual Entity Selection Capability

Under the current control interface, entity selection is handled through a list of active entities that is displayed in tabular form with the user using the mouse to make selections. This proved to be a limitation as there is no convenient way to correlate which entity is which within the visual scene. It would be a useful enhancement to select an entity within the visual scene for attachment without using this list. The visual application vendor is planning to add this feature, but has not set a date for when it will be available.
Bibliography


35
Appendix A - Operational Instructions

This appendix is intended to provided a working knowledge of how the TRIDIS Stealth is used. In the sections that follow, graphical representations of the actual windows and controls have been provided along with descriptions of how each control is utilized. All of the required script files listed in this appendix are located in the root Stealth directory (/usr/Stealth). For further information on software installation and directory structures, refer to Appendix B.

The complete Stealth installation will require the launch of two applications. The first application started is the DIS interface (section A.1) which will open the UDP port and configure the shared memory partition that will contain entity information and control structures. The second application is the visual application (section A.2) which contains the visual scene generator and the Visual Interface Manager.

A.1 DIS Interface Operations

Provided below are instructions that detail how to launch and configure the DIS interface that is used in the TRIDIS Stealth system. It should be noted that the described controls and capabilities do not represent a complete user manual for the DIS interface. Only those functions required to support the Stealth visual system have been discussed in detail.

A.1.1 System Configuration

Prior to starting the DIS interface, the Silicon Graphics system must be configured with the appropriate network settings. This should be left to the system administrator. The DIS interface will attempt to access the default EtherNet port utilizing the network address and broadcast mask set in the system configuration. Site and Host ID values will be prompted for when the interface is started, but these values are only reflected in the PDU header records, not in the UDP packet header. To set the SGI EtherNet port, the system administrator will need to access the SGI Irix ‘ifconfig’ utility and set the proper values. These values are site and exercise specific.

A.1.2 Starting the DIS Interface

The DIS interface is started by running the main executable ‘aiu’. The first thing the interface will do is prompt the user for the following information.

UDP Port Number : #
Exercise ID: #
Site Number: #
Host Number: #
The 'aiu' executable will then launch a series of executable tasks that constitute the DIS interface. Once launched, an x-terminal window will appear (see Figure A1) on the SGI main display device. This x-terminal contains a text-based menu set that allows the user to control the DIS processes, display status, and view an assortment of DIS debugging outputs.

**DIS Interface Startup Procedure:**

I. type 'aiu' <cr>

II. input the requested site, host, exercise ID and UDP port values.

The X-terminal window will appear. (see Figure A1)

![Figure A1. Main AIU Menu](image)

I.II. Select option 6 'Set Database Origin'

A menu listing the known databases will appear within the x-terminal. (see Figure A2)

I. Input the number that corresponds to the visual database that is going to be loaded.

The DIS interface should now be operating in the correct configuration. You may now proceed to start the Stealth visual system.
A.2.0 Visual System Operations

A.2.1 Configuring the EasyScene Visual Application

The EasyScene visual application is started through the invocation of a script file which identifies the database and task configuration that is to be used during the execution. A script file that corresponds to the provided database has been provided ('Stealth94.set'). For information regarding the preparation or alteration of these setup files, see the Coryphaeus EasyScene documentation. [CSI94]

A.2.2 Starting the Visual System

A startup script has been provided with the Stealth delivery package. The file is called ‘run_stealth’ and will set the appropriate environment variables and activate the visual system using the script mentioned in A.2.1. The file is located in the ‘/usr/Stealth’ directory and is the only command required to start the visual system. Within this file path information is provided to the visual task. Should any of the model or database files be moved from these locations, the new paths will need to be added within the script.
A.2.3 Visual System User Interface

The visual system has a GUI interface that allows a user to control various aspects of the Stealth operation. This control panel is switched on and off through the use of the F1 key on the Stealth systems’ keyboard. When active, the panel will appear along the left hand side of the main display screen. (see Figure A3)

In Figure A3 you will see the control panel with a brief description of what each block of controls is responsible for operating. This panel has been incorporated into the EasyScene executable with control input being processed once for each visual frame.

A.2.3.1 The Active Entity List

Shown in the top section of the panel, is a list of currently active entities along with descriptions of each entity type. Each entity is listed by site, host, and entity ID. To attach the Stealth eyepoint to a particular entity, the user must select the entity from the list by clicking on it. This list is updated every frame with the eyepoint moving to the next entry should the currently attached entity leave the network.
A.2.3.2 Eyepoint Storage

Figure A4 shows the eyepoint storage and retrieval control panel. These controls allow the user to store an eyepoint's location so that a particular view can be quickly restored when needed.

![Eyepoint Storage Table](image)

Figure A4. Eyepoint Storage Table

A.2.3.2.1 Storing an Eyepoint

To store an eyepoint in the list, the user needs to press the 'Add' button on the eyepoint storage and retrieval panel. This will bring up the 'Eyepoint Entry' control panel (see Figure A5). When displayed, the panel will contain the current coordinates of the Stealth eyepoint which can be viewed in any of the three coordinate systems indicated on the buttons at the lower left corner of the panel. At this point, the user should input a text descriptor for the new eyepoint record. Should the user wish to do so, different positional coordinates can be manually input at this time. Once complete, the “OK” button should be selected which will store the new record and close the panel.

A.2.3.2.2 Returning to a Stored Eyepoint

To return to a stored eyepoint, the user must select the eyepoint from the storage list with the mouse. A single click on the mouse button should be used.

A.2.3.2.3 Changing a Stored Eyepoint

To change the information in a stored eyepoint record, double click on the desired item in the eyepoint list. This will bring up the Eyepoint Entry panel discussed in section A.2.3.2.1. Modifications to the data fields will be saved when the “OK” button is triggered.
A.2.3.2.4 Storing and Retrieving Eyepoint Lists

When the need exists to store eyepoints between sessions, the stored eyepoint list can be written to a disk file by selecting the "Save" button in the eyepoint storage panel. The file generated will be named 'eyepoints.save'. This file is in an ASCII format and can be manually edited. To retrieve the stored eyepoints from disk, press the "Load" button and the "eyepoints.save" file will be read from the local directory.

A.2.3.3 Attachment Mode Selection

There are three attach modes currently supported by the Stealth package (Figure A6). The first mode causes the eyepoint to attach to whichever entity is selected in the entity list, at the top of the control panel. This corresponds to the Stealth Base Attach discussed in section 6.2.3.1. The second mode attaches the eyepoint to a stored eyepoint location in the Eyepoint Storage Table (Figure A4). The final mode is a free fly mode in which no attachment is made. The Eyepoint Attachment mode selector can be operated in two ways using the SGI mouse. The first method uses the left mouse button to step through each of the attachment modes and the second uses the right mouse button to activate a pull down menu from which the desired mode can be selected.
A.2.3.4 View Mode Selection

In conjunction with the Attach Modes discussed in A.2.3.3, there are three view modes available. The first of these view modes locks the eyepoint into a ‘Cockpit’ position for the attached entity (see section 6.2.3.1). The second mode is an observation mode in which the eyepoint is initialized in an over the shoulder ‘Observer’ or ‘Tethered’ position that can be altered using the controls discussed in section A.2.3.5. This mode is the most common ‘Stealth’ mode as it allows the viewer to follow an attached entity without being locked into a particular view angle. The final mode is a top down ‘Map’ mode. This is not to be confused with a plan view display. Rather, it provides a simple god’s eye view of the database that centers around the attach entity. This mode is for reference only and will place the visual system under maximum stress while in operation due to the large number of polygons required to render the entire database. The view mode selector can be operated in two ways using the SGI mouse. The first method uses the left mouse button to step through each of the attachment modes and the second uses the right mouse button to activate a pull down menu from which the desired mode can be selected.

A.2.3.5 Viewpoint Positioning

In Figure A6 the viewpoint positioning controls are shown. These controls operate in two modes. The first mode is a free fly mode in which the controls operate without regard for the movements of any other visual objects in the system. The second mode is the attached mode. While the visual eyepoint is attached to a DIS entity, the final eyepoint position and orientation is calculated as an offset from the coordinates of the attached entity. For example, when the eyepoint is attached to an entity, the eyepoint will initialize with a zero delta altitude thus following and matching the attached entity values. Should the user increase the altitude of the eyepoint while attached, the change is processed as a delta to the altitude of the attached entity, thus a 100 meter change would cause the eyepoint to always be 100 meters above or below that of the entity currently attached. This same principle applies to all of the position controls.

Note: When applying offsets to the attached eyepoint, it is easy to confused in terms of orientation and position. Within the viewpoint positioning controls is a ‘RESET’ button which will return the offsets to default values.

A.2.3.5.1 Lateral Movement

Lateral movement within the visual scene is accomplished by grabbing the ‘X-Y’ crosshair control and dragging it in the direction of desired movement. This is a continuous control and will affect the eyepoint position until released.
A.2.3.5.2 Elevation Control

Eyepoint elevation is manipulated through the use of the slider depicted in Figure A6. This is a continuous control and will affect the eyepoint elevation until released.

A.2.3.5.3 Pitch Control

Eyepoint pitch is manipulated through the use of a slider depicted in Figure A7. This is a static control and will remain as set by the user.

![Eyepoint Positioning Controls](image)

Figure A7 - Eyepoint Positioning Controls

A.2.3.5.4 Heading Control

Eyepoint heading is manipulated through the use of a dial that is depicted in Figure A6. This is a static control and will remain as set by the user.
A.2.3.5.5 Sensitivity Control

On the lower portion of the Viewpoint Positioning panel, there is a slider labeled ‘Sensitivity’. This control is used to control how sensitive the continuous controls (position & elevation) are to user input. If the user is trying to traverse the terrain rapidly, then this control will accelerate the process.

A.2.3.5.6 StealthPDU Transmission

As part of the preparations for the IITSEC 1994 testing and demonstrations, the facility to transmit Stealth PDUs that contain eyepoint information was added to the system. This facility is controlled with the ‘Transmit’ button within the viewpoint positioning controls.

A.2.3.6 Visual Scene Controls

In Figure A7 the visual scene controls are depicted. These controls allow the user to alter the visual scene characteristics.

Note: Under the current revision, weather effects are not implemented. The ‘Weather’ control switch is thus inactive.

A.2.3.6.1 Visibility Range

The visibility range is used to control the clipping mechanism that the visual system uses to determine how far away a terrain element can be while still being rendered. Use the slider control shown in Figure A8 to set this during run time.

A.2.3.6.2 Field of View/Zoom

As part of the visual scene control, there is an adjustment that allows the user to designate the FOV depicted in the visual system. Due to the way the field of view is processed, a broader field has the affect of zooming out of the visual scene. Normal operation for the Stealth assumes a 40 degree per screen FOV as the default configuration. It should be noted that wider FOVs impose a greater demand on the system as more polygons are processed into the scene.
Visual Scene Controls

![Visual Scene Controls](image)

**Note:** Weather Effects are not currently implemented.

Figure A8 - Visual Scene Control

### A.2.3.6.3 Time of Day

A simplistic time of day lighting model has been provided within the EasyScene system. Under the version used at I/ITSEC '94, this model simply allowed user control of ambient light levels from full daylight to zero light. This control is depicted in Figure A7. The slider control permits a time of day selection based on a 24 hour military style value.

### A.2.3.7 DIS Filtering Controls

The DIS filtering controls shown in Figure A9 are not currently integrated into the DIS interface. Control over the DIS filtering is currently handled via the DIS interface controls discussed in section A.1 of this document.

**DIS Filtering Controls**

![DIS Filtering Controls](image)

**Note:** The DIS Filtering controls shown have yet to be integrated with the DIS interface and are inoperative.

Figure A9 - DIS Filtering Controls
Appendix B - Installation Instructions

B.1.0 Stealth Installation Guide

This document provides all of the required information for the installation of the TRIDIS Stealth software. The directions that follow assume a Silicon Graphics workstation with the following software components installed.

- Irix Operating System (v5.2)
- Irix Development Option (v5.2)
- SGI Ansi C Language Compiler (v3.18)
- SGI Performer (v1.2)
- SGI GL Development Option (v5.2)

Step 1:
As the root user, 'cd' to the '/usr' directory and explode the archive file 'tridis_stealth.tar' with the tar utility ('tar -xvf'). This will create a Stealth directory with the structure shown in Figure B.1. The exact syntax for the tar command is system dependant. Check with your system administrator for needed device specifications.

```
Directory Tree for Stealth Installation

/usr/Stealth
  sgiAiu  VIM  build  Terrain Models
  \    /  |    |      /
  src h  src libs h

Figure B.1 - Directory Structure for Stealth Installation
```

Note: If execution only is desired, go directly to Step 3.
Step 2:
All of the makefiles reside in the build directory. (/usr/Stealth/build) These makefiles are configured by editing the ‘make.config’ file. Within this file, macros are defined to contain the paths to all compilation components. Should the user need to install the source code in other directories, the following macros should be changed to reflect the new directory paths.

AIU_HOME = /usr/Stealth/sgiAiu  *This path is for the DIS interface code.
VIS_HOME = /usr/Stealth/vim  *Path to the visual application code.

To rebuild the executable files, two Unix makefile scripts must be run by typing the following.

```
>makeAiuAll <cr>
>make -f makeStealth
```

The first script will build the DIS interface and the second will build the visual application.

Step 3:
Install Coryphaeus license server and product codes. These must be obtain directly from Coryphaeus Software. Although the code will compile without this license set, the visual application will not run without it.

B.2.0 Model Loading

The models used in the Stealth visual are stored in the ‘Models’ directory along with texture and support files. Control over which model is loaded for a particular entity type is handled in a text file (m.dat). This file, which is stored in the ‘/usr/Stealth’ directory, lists all of the recognized DIS entity type enumerations along with text descriptions and the names of the model files to load. The rules for building this file are as follows.

- List models in priority order.
- First match wins.
- Zeros are neutral place holders.

For example, the listings for a T80 tank might look like:

```
1 1 222 1 1 0 , T-80 Special Variant Tank, /Models/t80sp.drt,0
1 1 222 1 1 0 , T-80 Tank, /Models/t80.drt,0
```

The first record provides for a specific variant of the T-80 while the second record will provide a match for any T-80. To handle this situation, the records must be ordered as shown. When the list is scanned, the more specific entry will ‘hit’ only for the ‘specific’ type.
of T-80 and will ignore the ‘extra’ field in the type descriptor. If the second record is reached, it will cause any remaining T-80 variants to use the more generic model.

Note: This file can be changed and reloaded without restarting the visual system. Once the file has been edited, select the ‘Load Models’ button at the bottom of the control panel (Figure A3) and this list will be re-read and the proper models will re-load.

B.3.0 Terrain Database Loading

The terrain database that is loaded for the visual system is defined within the EasyScene setup file. For further information on the loading of different database variants, consult the EasyScene documentation provided by Coryphaeus Software with the run time license.
Appendix C - Vendor Information

C.1.0 Utilized Component Sources

DIS Interface Software:
SGI Application Interface Unit (June 1994 release)
Naval Research and Development
Navy Contact: Kevin Boner (619)553-3558

Visual Display Software:
EasyScene (v2.0)
Coryphaeus Software
12424 Research Parkway, ste 101
Orlando, FL 32826
(407)658-0041

Visual Models:
Viewpoint Data Labs
625 South State Street
Orem, Utah 84058
(800)229-3300

Terrain Database:
1994 I/ITSEC Hunter-Liggett Terrain Database
Silicon Graphics Inc.
SGI Contact: Graham Beasley (415)390-5420

C.2.0 Other Noted Vendors

Mak Technology
380 Green Street
Cambridge, MA 02139

Gemini Technology Corp
8 Pasteur Street
Irine, CA 92718

Computer Explorations, Inc.
917-A Willowbrook Drive
Huntsville, AL 35802
MultiGen
1884 The Alameda
San Jose, CA 95126

Paradigm Simulation, Inc.
15280 Addison Rd
Dallas, Texas 75248