ENHANCEMENTS TO THE DISTRIBUTED INTERACTIVE SIMULATION ARCHITECTURE FOR TRAINING SIMULATOR INTEROPERABILITY

Brett E. Butler
Loral Advanced Distributed Simulation
Orlando, FL
(407)384-3014, butler@wdl1.wdl.loral.com

ABSTRACT

The Strawman version of the Distributed Interactive Simulation (DIS) Architecture was unveiled in March 1992. This Architecture addressed the requirements and design of interactive (man-in-the-loop) combat simulation in a distributed and networked computing environment.

Since its initial unveiling, work has continued on the refinement and expansion of the architecture. This paper highlights developments which, as part of the Architecture, facilitate the interoperation of training simulators of varied fidelity, design, and manufacture.

A specific application which motivates this work is the requirement to conduct training simulation exercises which utilize three different classes of networked simulator devices—a class of existing DIS trainers, an existing high-definition engineering simulator, and a class of new-generation DIS simulators.

Specific issues to be addressed in the paper are:

- *Summary overview of key concepts of the DIS Architecture and its enhancements.*
- *Brief comparison of the configurations and capabilities of both the existing "SIMNET" devices and the newer-generation training devices.*
- *Analysis of how simulator differences detract from the interoperation of these systems.*
- *Discussion of the concepts and solutions, found in the DIS Architecture, which address interoperability problems.*

The focus of the paper is to address how the Architecture supports the implementation of interoperability solutions in the proposed exercises.

ABOUT THE AUTHOR

Brett Butler, a Systems Engineer with Loral Advanced Distributed Simulation, has over ten years of experience in the fields of flight simulation, networked simulation, and software engineering. Mr. Butler has an M.S. degree in Mathematics from Arizona State University.

INTRODUCTION

The Strawman version of the Distributed Interactive Simulation Architecture (known as the "DIS Architecture") was unveiled at the Sixth Workshop on Standards for the Interoperability of Defense Simulations in March 1992. This Architecture addressed the design requirements of interactive, man-in-the-loop combat simulation in a distributed, networked computing environment. The purpose of the Architecture is as stated:

> "The DIS Architecture defines a time and space coherent representation of a virtual battlefield environment, measured in terms of the human perception and behaviors of warfighters interacting in free play. It provides a structure by which independently developed systems *(e.g. training simulators)* may interact with each other in a well managed and validated combat simulation environment..."

Since its initial unveiling, the DIS Architecture has been expanded and revised. This work has been supported by the Army's Battlefield Distributed Simulation—Developmental (BDS-D) program. This paper highlights the key characteristics of the revised Architecture with respect to interoperability. Interoperability is the goal of facilitating the joint use of training simulators of varied fidelity, resolution, design and manufacture.

The specific application which motivates the work described in this paper is the requirement to conduct training simulation exercises which utilize three classes of simulator environments: an existing class of networked simulator devices (the Simulator Networking, or SIMNET system), a high-definition engineering simulator at NASA Ames Crewstation Research and Development Facility (CSRDF), and a class of newer-generation simulators being developed as part of BDS-D. This combined system will support tactical combat training—teaching the specifics of coordination, cooperation, and teamwork on the combined-arms battlefield.

Specific issues to be addressed in the paper are:

- *Comparison of the configurations and capabilities of the different simulator training devices.*
- *Analysis of how these differences cause problems in the proposed interoperation of these systems.*
- *Discussion of the solutions that have been developed to meet these interoperability problems.*
- *Demonstration of how these solutions are incorporated into the DIS Architecture.*

ARCHITECTURE OVERVIEW

The DIS Architecture was first presented in Strawman form in March of 1992. Since that time it has been expanded and refined. Space will not allow a full explanation of the Architecture. However, we will present some key features that relate to interoperability.

Architecture Composition

In general, an architecture consists of a reference model (to establish common conception and discourse), and an attendant set of standards (to establish commonalty in design and interoperability). In the past, the focus of the DIS community has been on the message protocol between simulators. This message standard (formally known as "Standard for Information Technology—Protocols for Distributed Interactive Simulation Applications" [7]) is commonly known as the DIS Protocol. DIS Architecture developers decided that the DIS protocol, while necessary, did not go far enough. Additional standards were needed to ensure compatible visual, terrain, weapons, and dynamics models. The Architecture uses the term "DIS Standards" for these additional standards specified by the Architecture.

Public vs. Private

Every architecture attempts to identify the boundary between private design and public conformance to a standard. The DIS Architecture is no exception. However, in DIS, with its emphasis on promoting interoperability between legacy simulation systems, this boundary line is even more important.

A tradeoff faced in this consideration is to weigh the promotion of comprehensive interoperability (by levying rigorous standards) against the accommodation of legacy systems. An architecture with light standards that accommodates legacy systems is said to be "non-invasive". As we shall see, the demands of useful interoperability under this application forced the architecture to be somewhat invasive.

Time and Space Coherence

Time and space coherence is the key objective of the DIS Architecture. Time and space coherence is not simulation fidelity. Fidelity describes how well the synthetic environment maps to reality. Time and space coherence is instead concerned with preservation of the simulation illusion and the maintenance of consistent experience and sensation for all simulation participants.

Implementation Principles

DIS technology is based on a core set of implementation principles which must be woven into the fabric of the architecture.

- *Autonomous simulation entities interacting in real time via networks using local copies of a common terrain and models database.*
- *Each DIS entity maintains its own world view, as a function of: its internal simulation, the common database, and state/event messages received from external entities.*
- *Each DIS entity employs Remote Entity Approximation (REA) to project a locally consistent time/space view of external entities.*
- *Simulation entities correspond closely to weapon systems and other actual equipment found in the synthetic environment.*

The last principle leads naturally to the creation of an object-oriented architecture.

Layered Architecture Model

The Strawman DIS Architecture focused on the physical implementation of networked simulation—the underpinnings of devices and networks which support the synthetic environment. The reference model has since been expanded to constitute a layered structure that gives greater expression to the synthetic environment and its algorithmic supports. The benefits of this layered model are fourfold:

- *layering simplifies module design in each layer*
- *greater emphasis on the synthetic environment supports applications and users*
- *Layer-to-layer assignment through the Architecture promotes independent and systematic exercise design*
- *Requirements trace through the Architecture layers supports enhanced Verification, Validation, and Accreditation (VV&A)*
- *Format emulation of other familiar architectural paradigms (e.g. OSI) supports acceptance and understanding of the architecture*

Figure 1 illustrates the layered reference model.

Synthetic Layer - This layer describes the synthetic environment in a hierarchical format, organized by classes and objects. It is used for descriptive purposes and provides benefit to fidelity description, VV&A, and expression of simulation requirements.

Logical Layer - This layer defines and describes entities and the underlying models of entity-to-entity communication. The format used is class and object descriptions which capture the methods and mechanisms of communication. This layer specifies entity types, allowable interactions between these types, and standard message formats

Physical Layer - This layer defines and describes the physical realization of DIS implementation. Its purpose is to describe: (1) implementation of entities, (2) implementation of interactions (including bit-wise design of PDU's and standard REA algorithms), (3) networking, (4) security, (5) interoperability, (6) strategies for VV&A, and (7) definition of "test-points" for testable, verifiable design.

Physical Layer Reference Model

The reference model for the physical layer is depicted in Figure 2. The essential components of this model are entities, cells, Cell Interface Units/Cell Adapter Units, and virtual networks. The diagram portrays three cells, one of which is shown in exploded view in order to depict its internal components.

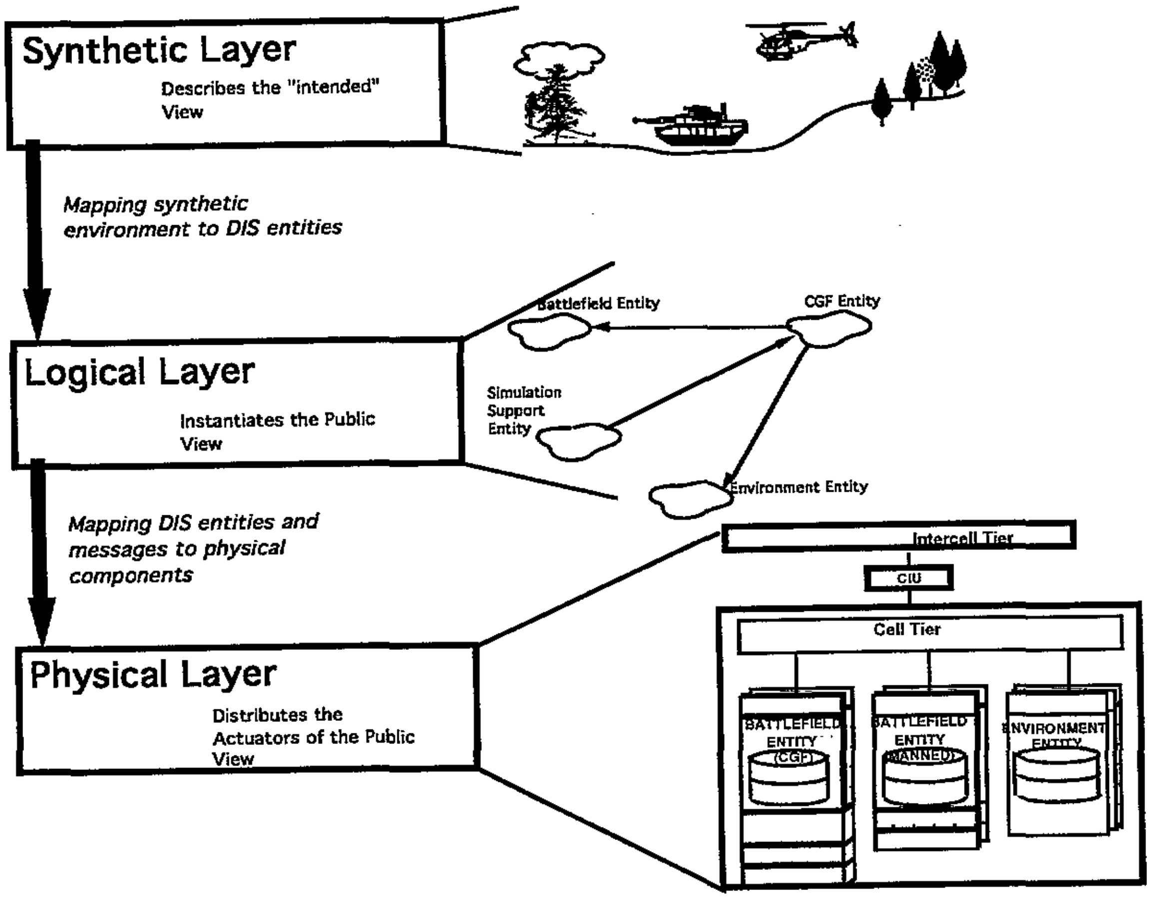

Figure1: Layered Architecture Model

Entities - Entities, as defined by the DIS Protocol are "elements of the synthetic environment that are created and controlled by a simulation application through the exchange of DIS PDU". The DIS Architecture broadens this terminology somewhat to describe simulation entities as objects that use the message-passing mechanism to interact with, *control, or monitor* the synthetic environment. The Architecture identifies three types of Simulation Entities: battlespace entities, simulation support entities, and environment entities. Battlespace and environment entities are properly "Simulation Entities" in the DIS protocol standard.

Battlespace entities correspond to actual battlespace equipment or organizations. They can be aircraft, ships, armored vehicles, dismounted infantry soldiers, guided missiles, command posts, trucks, lasers, emitters, platoon units, company units, etc. A battlespace entity incorporates a direct soldier/machine interface which emulates the soldier/machine interface associated with its real-world analog.

Simulation support entities "instrument" the simulation. They provide monitor and control, but have no analog on the actual battlefield. Examples include Plan View Display and Exercise Controller.

Environment entities corresponds to the components of the actual battlefield environment—terrain, atmosphere (haze, clouds, wind, etc.) bathysphere, sun lighting, moon lighting, and unmanned objects in the environment. They have no direct soldier/machine interface.

Asset vs. Entity - The Architecture uses the term "asset" to distinguish a simulation resource from its network presence. Entities, as described above, have network presence. As such, they can really be said to belong to all three layers of the Architecture. An asset is a physical resource (e.g. computer, cable, human operator, database) that supports an entity. To promote "plugable" interoperability, assets must be managed by the Architecture as well as entities.

Cells - Cells are collections of entities. Cells come in two varieties: Standard and Non-Standard. A Standard Cell contains only entities whose public parts conform to the DIS Architecture. A Non-Standard Cell does not. Non-Standard Cells may consist of non-DIS simulators, *instrumented live vehicles on a training* range, or analytic combat models.

All entities which share common databases and models and which are therefore "compatible" in the synthetic environment, belong in a common cell. DIS compliance, in the case of a Standard Cell, means that constituent entities communicate via the DIS protocol, and they draw their data from DIS-compliant Cell databases.

Figure 2 also describes the structure of the DIS Cell Database. This structure is known as the Common Database Standard (CDB) and is part of the Logical Layer. The DIS CDB consists of three component databases: Simworld Database, Session Database, and Review Database. The Simworld Database defines the underlying models of the synthetic environment (e.g. remote entity approximation algorithms, atmospheric models, terrain models, weapons and weapon effects models, rendering algorithms) and has a key role in supporting interoperability by ensuring commonalty of functionality among models and algorithms distributed across the network.

Virtual Networks - The virtual network connects entities for the *instantiation of the synthetic environment.* It has a two-tier structure. The cell-tier connects entities within a cell, the inter-cell tier connects cells. The Architecture specifies that the inter-cell tier must be DIS compliant. Linking the two tiers are devices known as Cell Interface Units (CIUs) and Cell Adapter Units (CAUs). *CIU's connect Standard Cells to the upper*

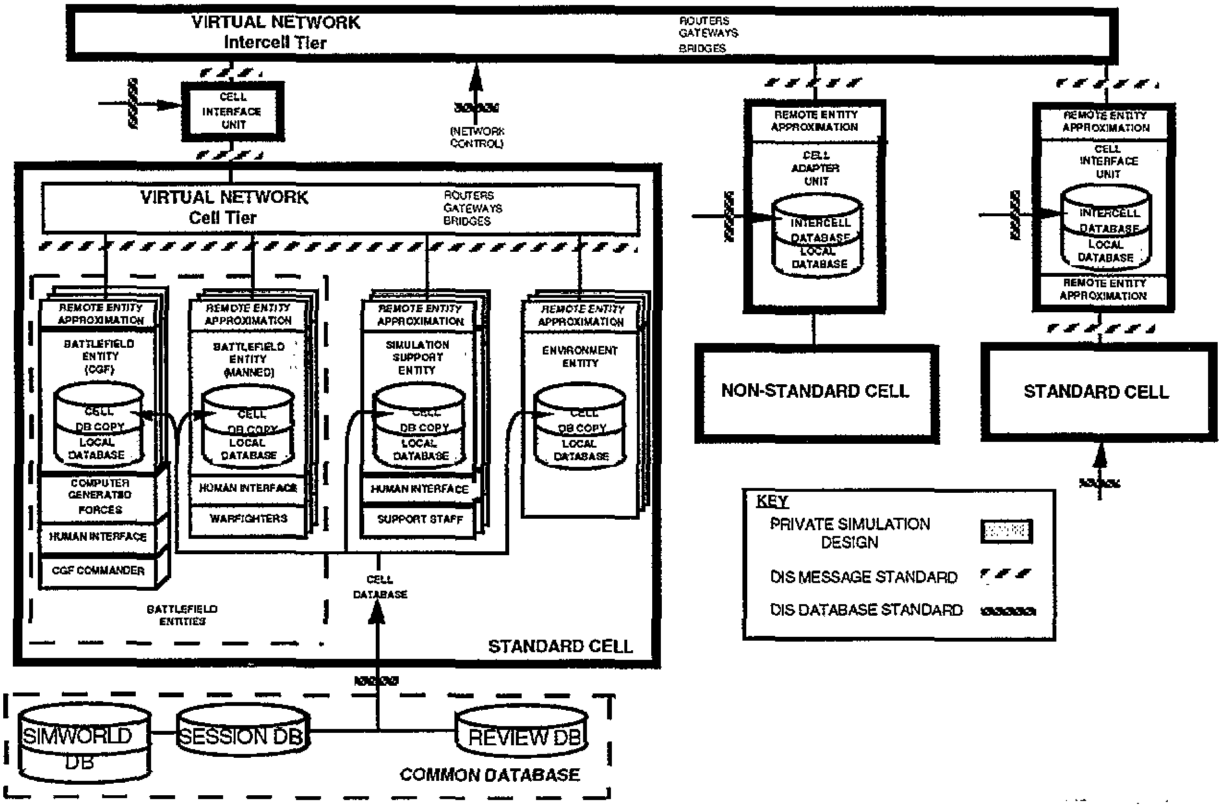

Figure 2: Reference Model for the Physical Layer

tier. CAU's connect (or adapt) Non-Standard cells to the upper tier.

Interoperability "Classes" and "Domains" — Two concepts in the Physical Layer *directly* address interoperability. The first, the Simworld Database, *identifies* "interoperability classes". The second, the cell, *contains* "interoperability domains". An "interoperability class" refers to a set of simulator (asset) characteristics which are sufficiently compatible in terms of algorithms, models, fidelity, resolution, database, security, throughput, physical connectivity, etc. to *validly* interoperate in most exercises. A cell contains a group of simulators, whose characteristics are all drawn from the same interoperability class, which are linked together to *meaningfully* interoperate. A cell is real assets linked together to support real exercises, hence an "interoperability domain".

Interactions between cells occur via CIUs and the CAUs. So here too the architecture addresses interoperability.

SITE DESCRIPTIONS

To sharpen its focus on the key issues of interoperability, we have concentrated Architecture extended development onto a BDS-D subsystem which highlights some of the relevant problems of interoperability. We call this subsystem the Architecture Design-Focus System (ADFS). A block diagram of the ADFS is presented in Figure 3.

The ADFS consists of three different cells (two existing, one proposed) that are being networked together to support joint exercises. Two cells are already connected: the Aviation Testbed facility (AVTB) at Ft. Rucker, AL, and the Crew Station Research and Development Facility (CSRDF) at NASA Ames Research Center in Mountain View, CA. The third cell (proposed) will be hosted at the AVTB site and will consist of upgraded flight simulators. In this paper, this new cell will be referred to as the "Level II Cell"— "Level II" being the designation applied to simulators in this cell, to distinguish their greater capability over the older "Level I" simulators of the AVTB.

Aviation Testbed (AVTB) Cell

Of the three cells, the Aviation Testbed (AVTB) has the most simulators in terms of both numbers and types. It represents mid-1980s technology, and was a key prototype system for DIS technology.

Connectivity - Simulators in the AVTB complex are linked via a standard 10 Mbps Ethernet. They communicate via the SIMNET (SIMulator NETworking) message protocol. The Ethernet is connected to a single long haul network by a gateway computer.

Simulators - The AVTB comprises several kinds of man-in-the-loop simulators. Among them are: generic rotary-wing aircraft (RWA), generic fixed-wing aircraft (FWA), M1 Tanks, M2/M3 Infantry Fighting Vehicles, and generic air defense devices (GADD's).

Image Generators (IG's) - For the aircraft simulators, visual imagery is generated through eight TV monitors by a dedicated IG. The IG outputs nine channels of video—eight low-resolution channels for out-the-window (OTW) visuals, and a single high-resolution channel for sensor simulation. The total field of view available is 125 degrees horizontal by 30 degrees vertical. The OTW views are vertically slewable and update in real time at a 15 Hz frame rate. The sensor views replicate day TV (DTV) and forward looking infrared (FLIR), and have various fields of view that are selectable by the CPO or CPG.

For the ground vehicle simulators, a dedicated IG generates eight low-resolution channels—seven are for vision blocks and one is for the gunner's primary sight (GPS).

Both types of IG systems are optimized for the display of moving models, a capability which figures prominently in tactical scenario simulation.

Tactical Environment - A system known as Semi-Automated Forces (SAFOR) represents enemy and auxiliary forces in the synthetic environment. SAFOR is a system of workstation-controlled computer generated forces that interact with manned simulators on the battlefield. Each workstation is capable of creating a battalion-size force. The purpose of SAFOR is to provide a larger battlefield context without being operator-intensive. SAFOR units are commanded by the Workstation operator who can execute pre-planned scenarios or create responses to evolving battlefield situations.

Level II Cell

The Level II Cell will consist of the latest-generation, modular, re-configurable man-in-the-loop simulators. These devices will be configured as generic RWA's, but the design will easily extend to other vehicle types. Plans call for a total of eight Level II simulators to be placed at this cell.

Connectivity - The Level II simulators will be connected by a private network, separate from the AVTB Ethernet. This network will utilize a state-of-the-art Fiber Distributed Data Interface (FDDI). Message communications over the FDDI link will conform to the DIS protocol.

Simulators - The Level II simulators will consist of state-of-the-art modular, reconfigurable cockpits. They will comply with the USAF-developed MODSIM architecture. Host computing devices will consist of distributed microprocessors running a real-time UNIX operating system.

Image Generators - Two candidate systems are under consideration for use with the level II simulators. Both systems represent the best of present-day, mid-priced IG technology. Common performance features between the two systems are high-resolution displays, fast update rates, and display of multiple moving models.

The first candidate system consists of three IG subsystems linked together by a high-bandwidth network. Through this network, the three systems share a common interface to the host computer for viewpoint and moving model control. Each of the IGs has access to the common terrain database making the three subsystems effectively operate as one IG. Two of the IGs are dedicated to OTW visuals. These subsystems drive three video channels each. The remaining subsystem drives two

The second candidate system was under development at the time of writing of this paper and design details are still sketchy.

Tactical Environment - The Level II cell will have a newer-generation SAFOR system known as ModSAF (for Modular SAFOR). ModSAF will have greater capability than the existing SAFOR system and will be able to simulate much of the sensor and EW phenomenology of the modern battlefield.

CSRDF Cell

The Crew Station Research and Development Facility (CSRDF) brings the highest fidelity of helicopter simulation to the ADFS. CSRDF is an engineering simulation designed to support helicopter

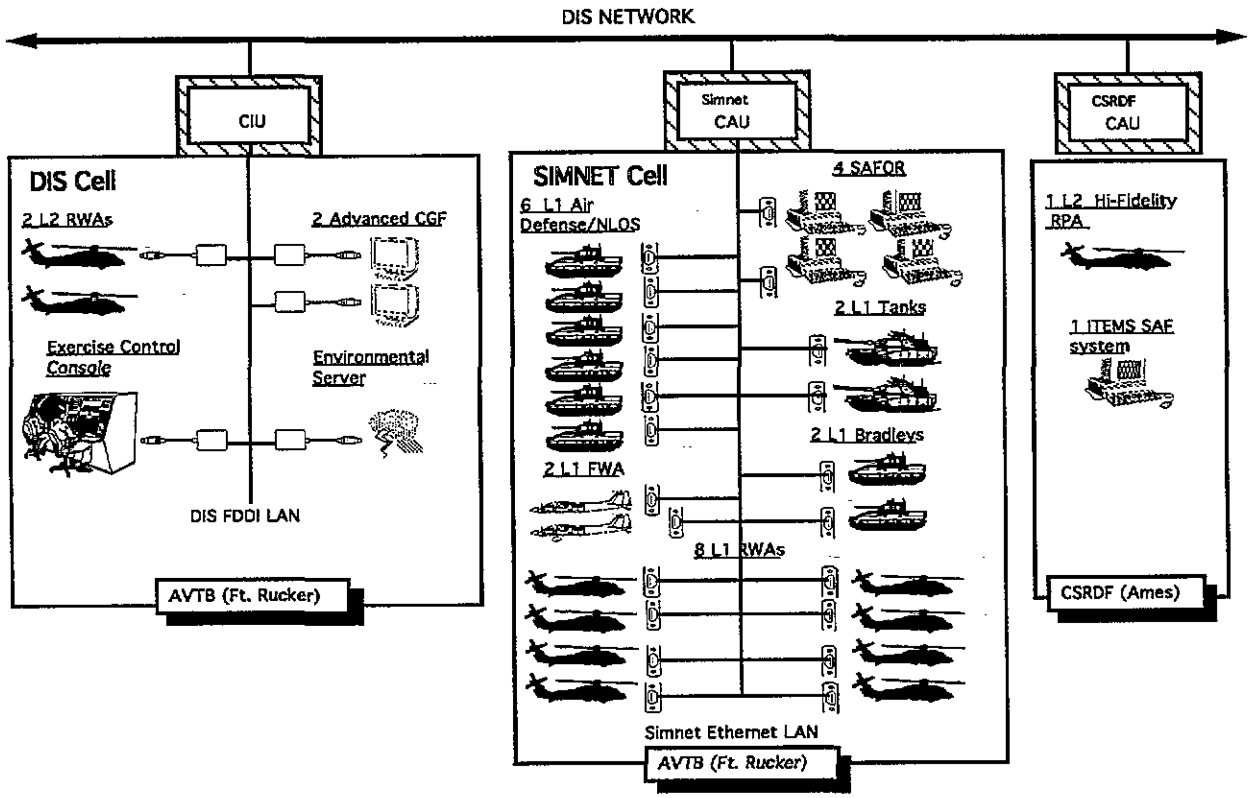

Figure 3: System Block Diagram (with Cells noted)

video channels and is dedicated to sensor simulation.

development. It therefore significantly differs from the tactical training simulations mentioned above.

Connectivity - The CSRDF facility is connected, via a long-haul gateway, with the Ft. Rucker AVTB facility.

Simulators - The CSRDF cell consists of one simulator configured as a two-seat cab on a fixed platform. It was designed originally to support evaluation of a 2-vs-1 crewmember question for the Army's LHX program, and was subsequently used to train LHX simulator assessment pilots prior to their visits to contractor sites. Since then, the CSRDF has been used for rotorcraft human factors research studies.

Image Generator - The CSRDF IG represents mid-1980's high-end technology. This four channel system is capable of displaying several modes: color OTW, FLIR, and DTV. A fiber optic helmet mounted display (FOHMD) is used to display visual information to the pilot. Of the four channels, one is used for each of the left and right background displays on the FOHMD, one for the high resolution inset display, and the remaining for the Automatic Target Recognition System.

Tactical Environment - The Interactive Tactical Environment Management System (ITEMS) is a software package being developed for CSRDF. ITEMS will be used for creation, control, and execution of the tactical scenarios with which the simulator crew will interact.

ITEMS will create and control all aspects of the tactical scenario: players (air and ground), player tactics, intelligent companions, adversaries, and gaming area weather. The capability of each player will be modeled to include maneuvering, active and passive sensors, weapons, signature and communications. Detailed modeling of guided, ballistic and static weapons will also be included.

INTEROPERABILITY CHALLENGES

Significant differences in simulation capability exist between the three ADFS cells. Integration of these systems, so that they can interoperate in meaningful training and research exercises, is contingent on solving problems that arise due of these differences.

Image Generator and Display System

Unquestionably, a major impact on interoperability is the differences in visual feedback received by trainees from their individual IG and display system. Due to the complexity of this subject, it is impractical to describe each of the four systems in detail. Instead, Table 1 summarizes those differences that can impede interoperability. Please note that capabilities described for each of the four systems are as procured for this application. System vendors may support additional capability through advanced products and enhanced options to the products described.

IG systems simulate both the OTW view and the sensor suite available to vehicle crews—Direct-View Optics (DVO), Day TV (DTV), Low-Light Level TV (LLLTV), and Forward-Looking Infrared (FLIR). In the paragraphs below, we offer additional explanation for the entries in the tables.

Update Rate - This characteristic describes how frequently the visual scene is updated or "re-painted" on the displays. Faster update rates lead to less scene jitter and more realistic, smooth motion.

Diurnal Effects Simulation - IG systems may simulate the changes in color and lighting that occur in the visual scene with the passing of the day (dawn, day, dusk, and night).

Visual Rendering Range - This characteristic describes the simulated viewing range at which an IG system can render visual objects. The comparison in Table 1 presents "best-case" rendering range, ignoring database density issues.

Moving Models - Because BDS-D is a tactical simulation, the number and complexity of moving models displayable is critical. Articulated components of moving models, such as rotating turrets, lend significant cues to warfighters. Moving

	L II (#1)	L II (#2)	AVTB	CSRDF
OTW Update Rate	30 Hz	30 Hz	15 Hz	60 Hz
Sensor Update Rate	60 Hz	60 Hz	15 Hz	60 Hz
Display Type	Dome	Dome	CRTs	FOHMD
OTW Resolution	4.17 arc min.	2.81 arc min.	4.13 arc min.	3.56 arc min.
Sensor Resolution	5.62 arc min.	5.62 arc min.	4.68 arc min.	3.56 arc min.
Diurnal Effects	Yes	Yes	No	Yes
Multiple Sensors	Yes	Yes	No	Yes
OTW Range	10 km	10 km	3.5 km	??
Sensor Range	15 km	15 km	7 km	??
Atmos. Attenuation	Yes	Yes	No	Yes
Total 6 DOF Models	100	74	64	16
Weapon Effects (3 DOF)	No Additional	40	128	32
Rockets/Tracers (3 DOF)	40	30	No additional	No additional
Polygons/Sec. - Total	396,000	360,000	210,000	240,000
Transport Delay - OTW	*100 ms*	*100 ms*	*167 ms*	*78 ms*

Table 1: IG and Display System Comparison

model display methods differ significantly between IG's and can be difficult to quantify.

Scene Density (System Output Performance) - The polygon display output of an IG determines the scene density that can be rendered. From a fidelity aspect, high density allows a closer representation of the detail of real visual scenes, providing the key motion and position cues required when flying close to the earth's surface. In applications like our aviation-oriented testbed, high scene density is critical.

Scene density is measured in terms of number of polygons displayed per unit time.

Resolution - Resolution is defined as the angle which is subtended by a pair of adjacent television raster lines on the image plane when measured from the design eye. The subtended angle is expressed in Arc Minutes of resolution.

Other IG Effects (Database and Load Management)- *When the density* of polygons in the database causes rendering overload, actions taken by the load management model can introduce unpredictable visual effects which can unbalance the fair fight. For example, IG "A" can be experiencing overload while IG "B" is not. Under normal loads each IG should render models at identical ranges, but in this example, IG "A" will pull in the rendering ranges of models, giving "B" a range advantage for target detection.

Occurrences of situations like this need to be kept to a minimum by utilizing good data base analysis and design practices.

Intervisibility

Pairwise-discrepant intervisibility between IG systems occurs when simulator A has clear line-of-sight to simulator B ,B has obstructed line-of-sight to A, and this discrepancy is due to an IG or database anomaly, not because of a reproduction of real-world features.

Miscorrelation between databases is one cause of this problem. Simulator A *may be intending to hide behind a rock or a* house that appears in its IG database. If the obstacle is positionally miscorrelated in B's database, the result may be that B can see A, A cannot see B (because of the obstacle), and A believes he is hidden from B. These problems can only be solved by a rigorous program of correlation.

IG-caused discrepant intervisibility is more problematic. Anomalies occur because of IG load management schemes. When confronted with more visual density than it can instantaneously process (too many polygons and too many moving models), an IG must adopt scene management techniques. Available strategies may include: not rendering distant terrain, dropping the level-of-detail of rendered models, and less frequent update of moving models. Any one of these strategies can cause discrepant intervisibility between scenario players.

Tactical Communications

Radio signal attenuation and distortion must be modeled so as to achieve uniform phenomenology in the synthetic environment. This problem is akin to the intervisibility problem.

The solution is to adapt a single-point "radio server" *on the network that* arbitrates all connection decisions by modeling attenuation and distortion.

Network Capabilities

Figure 3 portrays the network connections between the three cells of our design-focus system. The network configuration is highly mixed—local area network connections (LANs) of differing protocol, long-haul connections, and gateways. Differences in operational bandwidth between components of this network may impact interoperability.

Traffic on this network will consist mostly of the messages which describe entity positions and events. As additional entities join an exercise, network utilization proportionately increases. As network links begin to saturate, simulation entities begin to experience late message delivery and outright lost messages. Under extreme saturation, an affected link may fail entirely, dropping its dependent entities from the exercise.

Task Performance Fidelity

Differences in task performance fidelity may also affect interoperability via differences in warfighter workload. As the workload of the basic piloting tasks (flying, navigation, target recognition, acquisition, etc.) varies among the systems, simulation and training outcomes may vary with respect to what would occur in the real-world.

Tactical Environment

Differences in capacity and thinking ability of computer generated forces may also affect interoperability in terms of the fair fight.

INTEROPERABILITY AND ARCHITECTURE

Interoperability Types

The context of the Architecture, and its layered structure, allows us to clarify notions about interoperability.

A precursor to interoperability must be correct system and hardware interaction—the medium of communication between systems. These connections are documented and accounted for under the Physical Layer.

The initial threshold of interoperability is crossed by the implementation of the DIS Standards—the message protocol and supporting database standards. This implementation is described in the logical layer. This condition of meeting minimum interoperability requirements, we shall call *weak interoperability.*

Weak *interoperability* is not satisfactory for most simulation applications. We understand that interoperability can only be measured in the context of the synthetic environment. Starting with weak interoperability, one can proceed in two somewhat different directions in describing greater positions of interoperability. The first is a quantitative reckoning of the number of simulated functions that can interoperate. The second is a qualitative reckoning of the degree of interoperation versus real-world functionality. Complete quantitative interoperability we shall call *strict interoperability*. Complete qualitative interoperability, (an unattainable ideal) we shall call *strong interoperability*.

Strict interoperability means that the two systems interact appropriately in all modeled functions. For example, two different high-fidelity simulators may interact with each other in many areas of simulation (e.g.. sensors, visual appearance, navigation) yet not be strictly interoperable because of their implementation of different flight models levies different workload requirements on the subject pilots. Yet two different simple simulators, whose models do not account for much of the detail of the real world, may be strictly interoperable because of the inclusiveness of their interactions.

For systems that interoperate over many kinds of real-world functions, we shall say that they exhibit *strong interoperability*. There is no theoretical maximum to strong interoperability because of the inexhaustible amount of detail in the real world and our limited ability to represent it.

Two points can be made on this distinction.

(1) Strict interoperability and strong interoperability are not purely orthogonal concepts. For systems to be strongly interoperable, they must first share a high degree of strict interoperability.

(2) Strong interoperability is the most useful measure of interoperability for purposes of VV&A.

Figure 4 illustrates how the ADFS cells relate to these concepts.

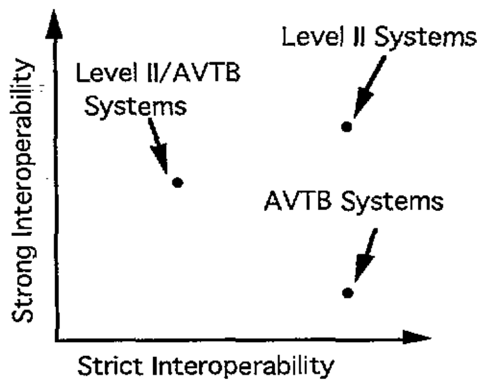

Figure 4: "Interoperability Space"

The "Fair Fight"

Interoperability directly affects the *fair fight*. When simulation is used for training or experimentation, the user intends to substitute simulation outcomes as surrogates for real-world outcomes. Hence the user tries to correlate simulation outcomes with outcomes in the real world. The term "fair fight" indicates that a strong correlation exists. The fair fight is related to strong interoperability in that it is evaluated by comparison to the real world.

There are two causes for the reduction of "fairness" in the fair fight during a simulation exercise: loss of time-space correlation and differing granularity in simulation fidelities.

Loss of time-space correlation can stem from miscorrelation among databases, "distribution effects", and simulator-specific differences. Miscorrelation of databases has already been discussed. Distribution effects are network-related anomalies such as latency, dropped packets, and out-of-sequence deliveries. These items can lead to a distortion of perceived time between two different systems in the exercise. Simulator-specific differences encompass database density and concomitant load management schemes.

Granularity in relative simulation fidelities can lead to differing workloads for pilots in the simulation.

CONCLUSIONS FOR THE ARCHITECTURE

In late 1992 and early 1993, an effort was undertaken to link two cells of the ADFS—the CSRDF cell and the AVTB cell. Initial requirements for this integration were not overly stringent. The key technical problems encountered involved: obtaining, converting, and matching terrain databases; and translation between two partially-disjoint message protocols.

Our experience with this integration effort, and with the design of the ADFS in general, generates valuable feedback for the future shape of the architecture in support of interoperability.

Validation of Need for Architecture Components

The most immediate results for the Architecture are the validation of the need for certain design components identified in the Physical Layer.

CAU—Because the AVTB utilizes the non-DIS-compliant SIMNET protocol, the first step toward the CSRDF-AVTB linkage was the development of a CAU to adapt the SIMNET protocol to the DIS protocol used on the long-haul network. A principal lesson learned is that the CAU must be of maximally high performance to sustain an exercise. Impediments to system throughput include the mathematically-intensive positional coordinate conversion routines. For this reason, project engineers experimented with substituting these routines with a polynomial-based mapping to promote greater speed.

Common Database (CDB)—The "handcrafting" of the terrain databases that had to occur to support interoperability shows the need for a strong CDB standard.

Environment Entities— These single-point arbiters of line-of-sight and radio connectivity are key to preserving the simulation illusion.

Reality of Exercise Design

The CSRDF-AVTB linkage relied on an obvious solution to the interoperability challenge—utilizing the lowest common capability between linked cells. This solution, "exercise design" (or downgrading), can be achieved by restricting exercise parameters to just those capabilities which can be systemically supported. Realistically, most future applications of DIS will have to rely on exercise downgrading to support interoperability.

The down side of this solution is obvious. Consistent application would require that, as long as there is at least one less-capable system in the exercise, all higher-capability simulators must play down to it. This strategy would neutralize the current investment in simulation equipment.

However, given the current situation, the Architecture should robustly support exercise downgrading. Current support consists of the means to capture and describe the reduced exercise capabilities through the class descriptions of the Synthetic Layer and the structures of the SIMWORLD Database. This descriptive capability promotes not only up-front exercise design, but also post-design validation and accreditation.

Given the importance of this area, developing the Architecture toward further support of exercise design would be beneficial.

Assert Interoperability at the Synthetic Layer

In discussion of Interoperability, attention must be focused on the phenomenology of the synthetic environment. Compliance to a standard message protocol is merely a precursor to interoperability. Meaningful interoperability can be assessed by use of the Synthetic Layer of the Architecture. To do so, one first identifies the intersection of the relative capabilities of the distinct simulation synthetic environments (as described by the Synthetic Layer class hierarchy). Then one identifies which parts of the intersection are implemented in the resulting combined synthetic environment.

Limitations of "Non-Invasive" Architecture

A "non-invasive" architecture (one that attempts to minimize required changes on legacy assets to have them play in DIS) will have limitations when it comes to supporting interoperability. Mere implementation of the message protocol is not enough to support full-bodied interoperability. One must be willing to modify legacy assets in order to support the degree of interoperability desired. A requirement for the Architecture, that stems from this realization, is that the Architecture must provide design guidance for such reconfiguration.

Floating Public-Private Boundaries

The *architectural* boundaries between the public and private parts of assets must not be considered a fixed line, but must be viewed as "floating" based on the degree of interoperability desired. The Architecture must support definition of this boundary.

ACKNOWLEDGMENTS

This paper has been supported by the U.S. Army's Simulation, Training, and Instrumentation Command (STRICOM) under the ADST contract.

The author would like to acknowledge the following individuals for their helpful input during the preparation of this paper: Bob Edwards, Steve McCarter, Dr. Richard Sherman, and Chuck Kuczaj.

REFERENCES

[1] Distributed Interactive Simulation Architecture Description Document. By Loral Systems Co. Orlando, FL: Simulation, Training, and Instrumentation Command, 1993.

[2] Distributed Interactive Simulation Common Database Standard. By Loral Systems Co. Orlando, FL: Simulation, Training, and Instrumentation Command, 1993.

[3] Kuczaj, Chuck. "ADST CSRDF Protocol Translator Integration & Test, Ft. Rucker Alabama, February 22, 1993— March 11, 1993." Trip Report, Loral Western Development Labs, 1993.

[4] *Hsu, J., Greenfield, B.* "The Fiber Distributed Data Interface Emerges As 100 Mpbs LAN Solution." Computer Technology Review. November, 1990.

[5] Rotorcraft Pilots Associate (RPA) Advanced Technology Transition Demonstration (ATTD). Appendix J. Crew Station Research and Development Facility (CSRDF). Author unknown. RPA bidder's conference handout, 1992.

[6] Sherman, Richard. "CSRDF Lessons Learned." Private Communication, Loral Western Development Labs, 1993.

[7] Standard for Information Technology— Protocols for Distributed Interactive Simulation Applications. Version 2.0. By The Institute for Simulation and Training, Orlando, FL. 1993.

[8] System Definition Document. By Loral Systems Company. Orlando FL: Simulation, Training, and Instrumentation Command, 1992.