(2)

ADST/WDL/TR-92-003010

AD-A255 249

Strawman Distributed Interactive Simulation Architecture Description Document Volume I: Summary Description

DTIC
S ELECTE AUG 10 1992 D
A

Loral Systems Company
ADST Program Office
Orlando, Florida

31 March 1992

Prepared for

PMTRADE
Program Manager - Training Devices
Orlando, Florida

92-21860

This document has been approved for public release and sale; its distribution is unlimited.

LORAL
Systems Company

92 8 6 033

ADST/WDL/TR--92-003010

Strawman Distributed Interactive Simulation Architecture Description Document Volume I: Summary Description

Loral Systems Company
ADST Program Office
12433 Research Parkway
Orlando, Florida 32826

DTIC QUALITY INSPECTED 3

Accesion For
NTIS CRA&I ☑
DTIC TAB ☐
Unannounced ☐
Justification
By per form 50
Distribution /
Availability Codes
Dist | Avail and / or Special
A-1

31 March 1992

Contract No. N61339-91-D-0001
CDRL A002

Prepared for

PMTRADE
Program Manager - Training Devices
Naval Training Systems Center
12350 Research Parkway
Orlando, Florida 32826-3275

LORAL
Systems Company

LORAL
Systems Company

ADST

REPORT DOCUMENTATION PAGE

Form approved OMB No. 0704-0188

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

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
03/31/92

3. REPORT TYPE AND DATES COVERED
Version 0
03/91 -03/92

4. TITLE AND SUBTITLE
Strawman Distributed Interactive Simulation Architecture
Description Document Volume I

5. FUNDING NUMBERS
Contract No. N61339-91-D-0001
CDRL A002

6. AUTHOR(S)
Beaver, Ray; Brasch, Randy; Burdick, Chuck; Butler, Brett; Downes-Martin, Stephen; Eller, Tim; Ferguson, Bob; Gotuby, Dave; Huber, Joe; Katz, Warren; Miller, Bill; Morrison, John; Sherman, Richard; Stanley, Walt; Yelowitz, K.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Loral Systems Company
ADST Program Office
12443 Research Parkway, Suite 303
Orlando, FL 32826

8. PERFORMING ORGANIZATION REPORT NUMBER
ADST/WDL/TR-92-003010

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Project Manager for Training Devices PM TRADE
Naval Training Systems Center
12350 Research Parkway
Orlando, FL 32826-3275

10. SPONSORING ORGANIZATION REPORT

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
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 may interact with each other in a well managed and validated combat simulation environment during all phases of the development process. A fundamental objective of the DIS architecture is to provide a blueprint to guide development of a general purpose system meeting the needs of a wide range of users. The architecture must therefore support the broadest possible range of user needs. At the same time, the architecture and the associated standards must provide sufficient design definition to achieve the goal of transparent interoperability of a very wide range of simulators, simulations, and actual equipment operating on instrumented ranges.

14. SUBJECT TERMS

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	17. SECURITY CLASSIFICATION OF THIS PAGE	17. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102

Table Of Contents

1 Introduction

This document presents a strawman Distributed Interactive Simulation (DIS) system architecture. It is a strawman in the sense that it is an initial architectural description, intended for review and comment by the community of DIS users, developers, and policy makers.

The focus of the document is on the definition of a system level reference model for DIS, and on the definition of the set of standards required to define and constrain the reference model to the extent necessary to ensure interoperability of independently developed simulators, simulations, and simulation exercises. The document also includes summary level discussion of the requirements which must be satisfied by the architecture, and presents in considerable detail the rationale leading to the strawman architecture and proposed standards.

Since the initial focus is on definition of the reference model and standards required to implement DIS, the document has been written for developers and users of DIS simulations. As the document and its supporting standards mature, it should provide the framework for design and development of new DIS simulations. It will also provide a useful reference for newcomers to the DIS community by making explicit the underlying principles of DIS.

The architecture draws very heavily from the body work in process by the DIS Conference for the Interoperability of Defense Simulation, which is jointly sponsored by the Army Program Manager for Training Devices (PM TRADE) and the Defense Advanced Research Project Agency (DARPA). Under the auspices of the DIS Conference, representatives of the military services, industry, and associated research organizations have been working toward definition of an industry standard for Protocol Data Unit (PDU) messages. These PDU message provide the basic means of interaction between DIS simulation entities. The architecture described by this document is generally consistent with the work underway by the conference, but attempts to provide a capstone document which defines the context for the work underway. However, in some areas the strawman architecture extends beyond the work of the conference by proposing establishment of a specific reference model context for the PDU Standard, definition of some standard terminology for the components of the reference model, and creation of an additional standard governing the set of databases required to support DIS exercises.

The document is organized as two volumes. Volume I presents the strawman architecture and briefly address the major issues associated with the architecture. Volume II (consisting of two books) presents supporting rationale and addresses some of the fundamental issues underlying DIS and the strawman architecture.

Comments and recommendations for changes and improvements are welcomed and encouraged. Comments may be submitted to:

Loral ADST Program Office
12443 Research Parkway
Suite 303
Orlando FL 32826
attn: DIS Architecture

Comments may also be submitted via the ADST Bulletin Board System(BBS). Post comments in the DIS Architecture Comments area of the BBS. This area is found within the following structure:

ADST Bulletin Board System
System Description and Technical Information
DIS Architecture
DIS Architecture Comments

If you are not already a registered user of the ADST BBS, send a registration request to:

Loral ADST Program Office
12443 Research Parkway
Suite 303
Orlando FL 32826
attn: BBS Administration

2 An Overview Of DIS

2.1 Vision:

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 may interact with each other in a well managed and validated combat simulation environment during all phases of the development process.

2.2 Objectives

A fundamental objective of the DIS architecture is to provide a blueprint to guide development of a general purpose simulation system meeting the needs of a wide range of users. The architecture must therefore support the broadest possible range of user needs. At the same time, the architecture and the associated standards must provide sufficient design definition to achieve the goal of transparent interoperability of a very wide range of simulators, simulations, and actual equipment operating on instrumented ranges.

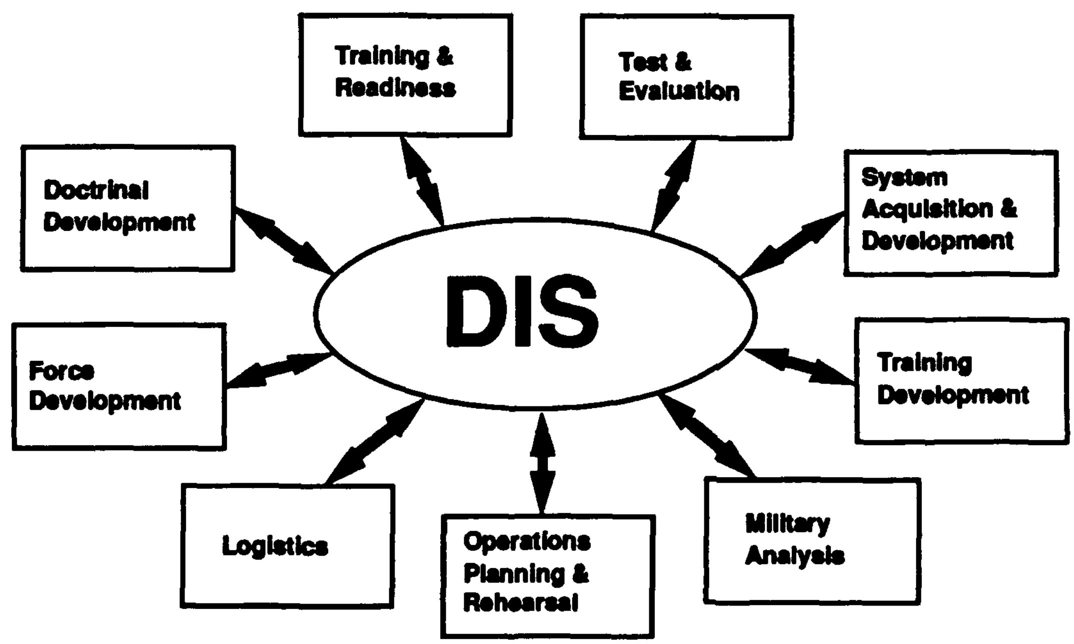


Figure 2.2-1: The Electronic Battlefield must meet the diverse needs of military users in all stages of the development process.

The following sections define the top level user objectives that the DIS architecture must serve, and the top level design principles which guide the architecture definition, in order to meet the needs of this diverse group of users.

2.2.1 User Objectives and Requirements

Given the diversity of users and the broad potential impact of DIS, as well as the experimental and innovative nature of DIS, it is impossible to list a definitive set of user requirements that will ensure evolution of a single fully integrated DIS system addressing all user community potential needs. Likewise, since DIS is a new and rapidly evolving capability, great care must be exercised to impose only the minimum set of design constraints necessary to achieve sufficient interoperability without imposing restrictions and constraints on growth and creativity. However, in order to define an architecture, a set of basic objectives and design principles must be defined to guide the development of the architecture. The following is a summary of those objectives and design principles. Some of the major implications of each objective and principle are also noted.

One result of the diversity of user communities served by DIS is lack of a common vocabulary. To avoid confusion, the terminology used throughout this document must be interpreted using definitions specific to DIS. Appendix A of this volume provides definitions which clarify the meaning of the key terms. Appendix A should be reviewed by the first time reader since certain common terms have distinctive meanings in the DIS context.

User Objectives Summary

- *The architecture is intended to support simulation needs throughout all phases of the development cycle.*

  Development in this context encompasses all aspects of combat development, materiel development, testing, training, and mission rehearsal. This is a fundamental requirement for DIS. Because of the diversity of needs throughout the development process and application to all aspects of the battlefield, the architecture must allow simulation of a wide range of battlefield interactions, at various levels of fidelity, to be conducted in a common battlefield environment.

- *The architecture focuses on warfighter-in-the-loop simulation of collective team battlefield interaction.*

  Exercises will typically include multiple manned simulators or man-in-the loop simulations. This implies that the minimum thresholds for fidelity levels, rates of interaction, etc. can be defined with respect to the subjective reference "sufficient to engender realistic soldier behaviors and interactions."

- *The architecture must support free-play exercises, but allow for controller interactions.*

  In the absence of controller inputs, all outcomes on the DIS battlefield are the result of interactions between the participating entities, such that cause and effect relationships are maintained. No controllers are required on the DIS battlefield to resolve outcomes, but controller intervention is supported.

- *The architecture must support exercises consisting of multiple heterogeneous simulation entities.*

  Many different types of simulation entities must be supported, including simulators of varying fidelity, networks of simulators and actual equipment operating on instrumented ranges, stimulated actual equipment, and mixes of simulators and simulations. The DIS architecture must also provide for backward compatibility to support interoperation with all types of existing simulator and simulation assets. While it will be possible for any DIS-compliant simulation entity to participate in exercises on any DIS battlefield, the degree of simulation validity which results may not satisfy the intended purpose. It is the responsibility of the architecture to provide a framework which identifies these differences in simulation validity. It is the responsibility of the user to assess the validity of the exercise in light of these differences and the exercise objectives.

- *The architecture must support exercises consisting of highly geographically dispersed simulation entities.*

  DIS brings simulation to the user, not vice versa. Worldwide geographic dispersion is a DIS goal.

- *The architecture must support exercises ranging in scale up to thousands of simulation entities at hundreds of sites.*

  An initial goal is system capacity to support thousands of simulation entities within a DIS exercise by the mid 1990s. Future expansion is envisioned as higher capacity networks and more powerful computer systems become available. The architecture avoids placing limitations on the number or type of entities that can participate in an exercise.

- *The architecture must support multiple, simultaneous, independent exercises, including multiple classified exercises as well as unclassified exercises.*

  Multiple, independent, simultaneous virtual battlefields, including battlefields at various levels of fidelity will be required to serve the diverse and independent needs of the DIS community of users. Likewise, battlefields must be supported which represent various time frames, including historic, current, and postulated futures. Multiple levels of security are required so that some exercises can operate with data classified at levels up to SECRET. Higher levels of classification can also utilize the DIS architecture, but such exercises would normally be conducted using dedicated secure facilities.

- *The architecture must support efficient validation of exercises, simulation entities, and supporting databases and models.*

  Validation, however, is not a function of the architecture but rather of the standards and supporting infrastructure which the architecture identifies. This requires that effective configuration control can be maintained over all elements of the architecture, and that authorized users have full and easy access to all DIS parameters, models, and databases.

- *The architecture must support efficient reconfiguration of simulation entities to create exercises with new combinations of entities.*

  Standard interfaces are necessary but not sufficient to ensure reconfigurability. In addition, efficient means must be provided to allocate assets to exercises, share databases, and determine the validity of the result. To the extent possible, means must also be provided to compensate for differences in fidelity and/or capability when heterogeneous simulation entities participate in an exercise.

- *The architecture must support efficient development of new/modified simulation entities.*

  This requires that users have access to copies of all parameters, models, and databases and can modify these copies to adjust performance, create new entities, and perform experiments, while maintaining visibility of validity compromises. It also implies an implementation which permits and encourages reusability of software, data, and developmental hardware.

- *The architecture must support applications incorporating multiple Service echelons, from small unit to theater.*

  One of the promises of DIS is the ability to create exercises in which all levels of the command hierarchy can interact in a realistic battlefield command and control environment. Current simulation capabilities provide only rare opportunities for such interaction and engender large costs.

- *The architecture must support all Services.*

  The architecture is intended to fully support joint and allied exercises.

- *The architecture must support the use of semi-automated and automated computer generated forces to expand the number of entities on the simulated battlefield.*

  The Computer Generated Forces (CGF) provide both supporting and enemy forces indistinguishable from fully crewed simulation entities. The use of CGF reduces the simulation resources and manpower needed to conduct realistic battles, to represent platforms and weapon systems for which no manned systems exists, and to reduce or eliminate requirements for role playing actors to participate in battles.

- *The architecture must support flexible comprehensive exercise instrumentation and collection of exercise data, including accurate replay of exercises.*

  One of the most powerful attributes of DIS as compared to all other forms of simulated engagement is an inherent ability to collect accurate and comprehensive exercise data by capturing the message data which defines all entity interaction. Ability to replay exercises depends on such comprehensive data capture, and is postulated as a basic requirement for DIS. Requirements for instrumentation vary greatly depending on the purpose of the exercise. Because of the need to serve multiple communities of interest, the DIS architecture provides maximum instrumentation flexibility. This includes the ability to extend the inherent capability of DIS by making provision for insertion of custom data probes into simulation entity internals to capture internal state data along with the basic DIS PDU message data.

- *The architecture must support exercise control functions including exercise reset/restart.*

  Practical implementation of multi-site exercises requires standardized exercise control functions. Reset/restart requirements imply requirements for capture and logging of simulation internal state data in addition to the data captured by recording PDU message traffic. Replay/restart therefore places implementation requirements on the "private" internal components of simulations.

- *The architecture must provide high system availability.*

  Fault tolerance is a basic requirement of the DIS architecture. Single points of failure which can halt an exercise must be minimized, and exercises must be fault tolerant, in the sense that failure of a simulation entity or loss of a PDU message must not cause the exercise to fail.

2.2.2 Implementation Principles

In addition to the user objectives described above, DIS implementation principles can be defined. These principles are generally transparent to the user and are primarily of interest to the developers of DIS compliant systems and DIS standards, and to those responsible for determining the validity of DIS exercises.

DIS is a direct descendent of SIMNET and has inherited most of its implementation principles from SIMNET. However, there are significant differences. Since DIS closely overlaps SIMNET, the list which follows points out the similarities and differences.

DIS Implementation Principles Inherited from SIMNET:

These principles are closely related and must be considered as a group.

- *DIS consists of autonomous simulation entities interacting in real or wallclock time via networks using local copies of a common terrain and models database.*

  In this context, autonomy is intended to mean a distributed computing environment in which each entity is equipped with all required processing resources, such that entities can leave or join exercises independently. It also means that simulation control is distributed among autonomous simulation entities. Autonomy is important to reduce vulnerability to single-point failures, to promote easy reconfiguration of assets into exercises, and to allow continued growth without requiring extensive changes throughout the system.

  The two key elements of DIS are the message standards that permit interoperability among autonomous simulation entities and the common terrain, weather, and models of physical events which the individual simulators each maintain in local data bases in order to interpret the inter-entity messages.

- *Each DIS entity maintains its own world view, based on its own simulation, the common database, and the state/event messages received from external entities.*

  No attempt is made to synchronize all world views from all simulators. This means that it is possible for simultaneous events to occur in different orders depending on where they are perceived from. Only entities within another entity's sphere of perception are presented on that entity's visual, radar, or other sensor display.

  There is no need to maintain a history of the exercise at each simulator. Participants joining or rejoining an exercise will be updated on the location of all other exercise participants in a matter of a few seconds by messages arriving from them.

- *Each DIS entity employs Remote Entity Approximation (REA) to project a locally consistent time / space view of external entities.*

This is often called "dead reckoning" or "remote vehicle approximation". Dead reckoning implies first order (velocity only) projection, rather than the use of higher order projection algorithms supported by DIS. Remote Vehicle Approximation avoids that limitation, but DIS entities may not be vehicles in the usual sense. Consequently, this document uses the term "remote entity approximation" or REA.

Use of remote entity approximation is one of the most basic implementation principles of DIS, in that it makes possible wide geographic dispersion and support for large numbers of entities by greatly reducing inter-entity message traffic and network bandwidth requirements, and by relaxing requirements for frame synchronization between simulation entities. This implementation benefit, however, requires tradeoffs between bandwidth and degree of time/space correlation maintained between participating simulations. Employment of remote entity approximation also trades network bandwidth for increased processing workload for each simulation entity, in that each entity must interpolate entity locations at the simulation computational frame rate. The computational load becomes significant under several conditions: as the number of entities in an exercise grows, as higher order algorithms are employed, and as smoothing techniques are employed to reduce visible vehicle movement anomalies caused by approximation errors. Thus, the limits of human perception enter into the trade off between computer power, network capacity, and visual fidelity. Entity dynamic capabilities and characteristics (speed, acceleration, etc.) and types of interaction between entities (station keeping requirements, collision bounds, etc.) impact the error tolerance, and therefore enter into these trades as well.

Volume II of this document presents an analysis and discussion of these tradeoffs.

- *Simulation entities correspond closely to weapon systems and other actual equipment found on the battlefield.*

"Actual" includes conceptual battlefield equipment as well as current or historic equipment.

This principle is required to allow efficient validation of exercises, since entity interactions must correspond to actual battlefield entity interactions. Therefore cause and effect relationships in the simulation correspond to cause and effect relationships in the real world. An underlying thrust of this principle is to enhance the realism of the DIS simulators and simulations by better replicating military operations and the interaction of battlefield entities with each other and the environment.

This principle also supports the objective of easy reconfiguration of exercises with new combinations of simulation entities to represent new combinations of battlefield entities, and addition of these new battlefield entities to exercises.

Additional DIS Implementation Principles:

- *The architecture must expand the concept of networked simulation to include a wide range of heterogeneous simulators and simulations.*

  SIMNET involved a relatively small set of essentially homogeneous simulators. DIS expands considerably on this limited set and consequently encounters serious concerns over the fidelity, validity, and interoperability of the entities participating in an exercise. At the present time, no general solution exists to the challenge of making heterogeneous simulators and simulations interoperable.

- *The architecture must support and encourage reuse of software, while avoiding obsolescence by promoting continued improvements and additions.*

  Realization of this principle requires an architecture which supports object oriented software, and creation of a software library infrastructure to support simulation entity developers and maintainers. Documentation standards must also be established for such software at levels sufficient to meet the reuse objective. The software in the library could include public domain software, government owned software, and commercial software available through license.

  This principle raises some policy issues which are discussed in Chapter 5 of this document.

- *Openness is a primary objective of the architecture.*

  Openness requires that all information required to design new simulation entities or to modify existing entities is readily accessible. To the maximum extent practical, industry standards should be used in order to maximize application of commercial off-the-shelf products and the base of practical technical knowledge that surrounds such products and standards. Additional standards, specific to DIS, will be developed by the DIS community to promote easy creation of new battlefields, new exercises, and new simulation entities. These standards will be presented to the community of DIS users and designers, reviewed, and, if accepted, submitted to a recognized standards group, such as IEEE.

2.3 DIS Regime

The user objectives and implementation principles stated above lead to global requirements that must be supported by the DIS architecture. The most pervasive and general requirement is focus on man-in-the-loop simulation. Man-in-the-loop implies many things, but a primary aspect is simulation of battlefield interaction between multiple warfighters at levels of fidelity sufficient to invoke realistic decision making behavior by the participants. This focus leads to a general DIS regime of suitability. There may be reason to use DIS outside this regime, but the design tradeoffs have been optimized for use within the regime, and stated levels of simulation capability generally don't apply outside this regime.

Since the focus of DIS is on warfighter interaction, one characteristic of the DIS regime is use of real-time as simulation time. It is possible to envision some higher command level interaction that realistically can be conducted using simulation time faster than real-time, but to the extent that a simulation includes platform level simulators and/or actual equipment, real-time must be used. Of course, it is possible to jump in time, but such time shifts correspond to cessation of one simulation and start of another at a new simulated time; no continuity is maintained over the jump.

Another characteristic of DIS simulation is maximum entity interaction rate. Simulation of interactions which occur between entities in the real world at a given interaction often requires interaction at or above that rate. In keeping with the DIS focus on human perception, DIS is designed to support interactions between simulation entities at rates sufficient to maintain the illusion of reality at the limits of human perception. The upper limit on interaction rate is set by the end-to-end delay from a warfighter action in one entity to the display of that action to another warfighter in another entity. Since most warfighter-in-the-loop simulation entities include out-the-window view Image Generators, and because DIS must support interaction between widely dispersed entities over long haul networks, the upper limit of interaction rate is limited by simulation processing, image generation, and long haul network delays. By carefully controlling these delays, interaction rates approaching the limits of perception can be achieved, but these delays establish the boundary of the DIS regime.

One result of this interaction rate is a limitation on the simulation entity level within the DIS regime. DIS implementation principles state that DIS entities should correspond to actual battlefield entities, but battlefield entities can be defined at a wide range of aggregation level. Table 2.3-1 roughly defines some entity levels by example.

Table 2.3-1: Entity Levels

Entity Level	Example
Part	switch, spark plug, target seeker, gun barrel...
Module, Component	sensor, weapon system, propulsion system, ...
Platform	tank, truck , aircraft, guided missile, dismounted infantryman, SAM site...
Unit	squad, platoon, company, battalion, brigade, ... flight; battlegroup; ...

The current DIS PDU standard focuses on platform level simulation entities. A basic characteristic of such platform level entities is that they can be portrayed visually as a rigid body with a single location and orientation, or as such a rigid body plus some attached articulated (moving) parts. Platforms which contain other separable entities (such as TOW launchers, bombers or submarines with cruise missiles, or helicopters ferrying troops) are also supported.

Higher echelon commanders often do not have a direct view of the battlefield, seeing it only as maps, symbols, reports, etc. Therefore, simulations for higher echelon commanders could be implemented with DIS using either platform level entities which are reported as aggregated unit level entities, or as actual unit level entities. Unit level interactions are generally slow (use large time steps) compared to platform interactions, so that the interaction rates provided by DIS are more than adequate for unit level simulation.

The obvious limitation, however, is that unit level simulation cannot represent platform level interactions at full fidelity nor interact directly with platform level simulations. The platform level simulators must simulate visual or other sensor interaction with other platforms rather than with units. Unless the unit level entities are deaggregated, displayed, and interact as platforms, they cannot interact with platform level entities. Furthermore, since actual battlefield interaction takes place at the platform level, cause and effect relationships can be validated most directly at the platform level. This leads to consideration of a policy to focus all DIS interaction to the platform level.

The DIS architecture is not designed to support interaction between distributed but tightly coupled modules interacting at high data rates, as is typical of interaction between modules within a single platform (e.g. interaction between a missile guidance system and the missile dynamics). DIS long haul network latencies preclude the required levels of interaction. The phase lag in the networked simulation feedback loop is far too great to simulate the actual equipment feedback loop. Simulation of such a tightly coupled system therefore requires a centralized computing environment or a very high speed local area network. The DIS message protocol could be used within a centralized system, but it is not designed or optimized for that purpose. At least two other DoD sponsored simulation standards efforts, the Modular Simulation (MODSIM) Program and the Joint Modeling and Simulation System (JMASS), are underway to develop interface standards for component-level simulator and simulation entities.

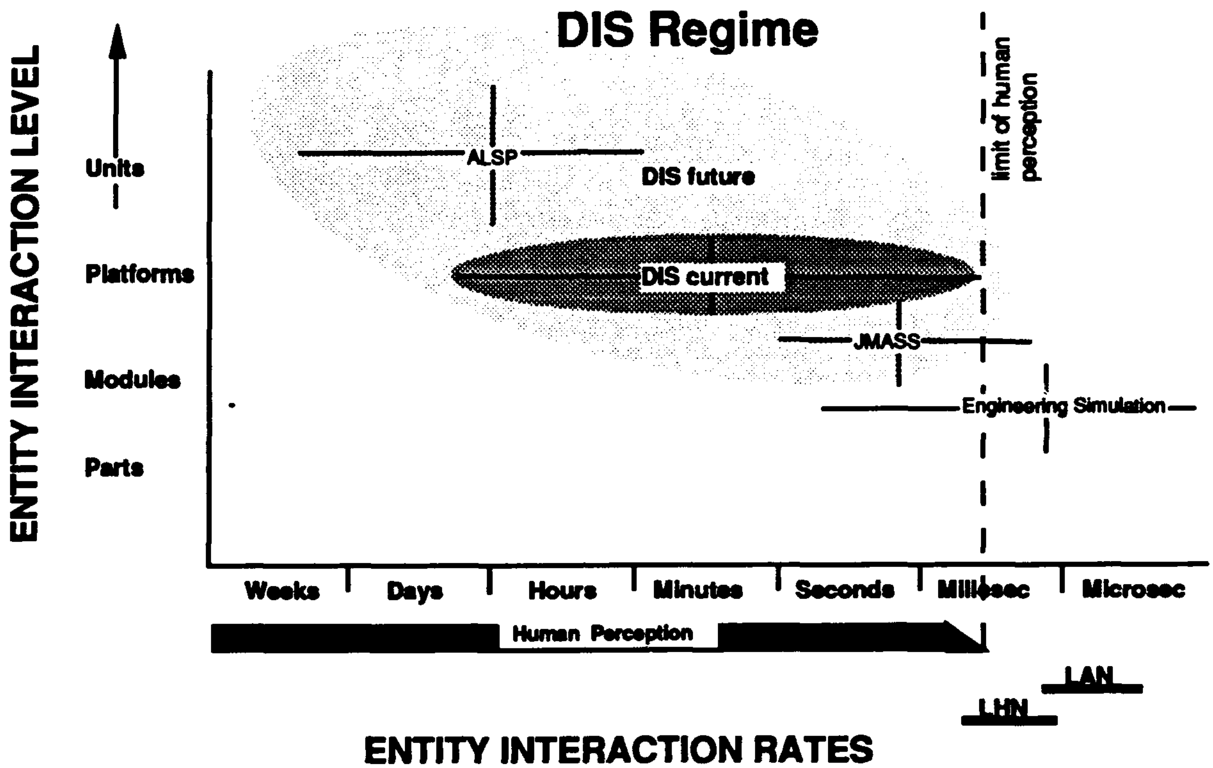


Figure 2.3-1: The DIS Regime.

The DIS Regime viewed in terms of entity level and entity interaction rate is illustrated in Figure 2.3-1. The figure also shows the approximate regimes of some other well-known simulations and simulation types such as the Aggregate Level Simulation Protocol (ALSP), MODSIM, and JMASS. Additional information on each of them can be found in Chapter 4.

The issue of time/space coherence and human perception in DIS is discussed in greater detail in Volume II of this report. Unit level entity interaction leads to discussion of the relationship of DIS to High Order Models and to modeling and simulation of command and control structures as a key component of Computer Generated Forces. Both topics are also addressed in Volume II.

3 Architecture Description

The strawman architecture is described in terms of a reference model which helps define terms and relationships between the elements of the logical architecture, and a set of standards which apply to the elements and interfaces of the reference model. The reference model divides the overall DIS network into a small set of DIS elements sufficient to describe the architecture.

Section 3 discusses the basic attributes of DIS and defines the set of elements which comprise the DIS reference model. It then proposes a set of standards to be applied to the reference model to define a DIS architecture. Examples of specific DIS implementations are then presented as illustrations of the architecture in practical terms.

Please note that terminology used throughout this document is intended to comply with the definitions in the Glossary of Terms in this Volume. DIS cuts across many user communities and technical communities, each with its own standard terminology. The writers frequently found that terms used had slightly different meanings in different communities. Every attempt has been made to use terms consistently and in accordance with the glossary throughout this text. In particular, note that the term "entity" is synonomous with the term "simulation entity" unless otherwise noted. An entity is a simulation with a specific DIS interface. Later in this discussion, various entity types are defined. A node is a physical implementation of one or more entities. A processing node is a computer system, and a network node is a discrete interface to a network. The terms "element" and "component" are used in the normal "part of" sense.

3.1 DIS Design Principles

The basic elements of the DIS reference model are simulation entities and an interconnecting network. The DIS architecture and associated standards must define entity interfaces and interactions with sufficient precision that developers with knowledge only of the reference model and the appropriate standards can create new modules capable of interaction with the other simulation entities.

A basic premise of DIS requires that each DIS asset is to the maximum extent practical self-contained and autonomous, such that new assets can be added and existing assets can be reconfigured with minimum impact on other assets. This is achieved by building the system using multiple discrete largely autonomous entities. This encourages (but does not require) assets to be packaged as discrete independent processing nodes.

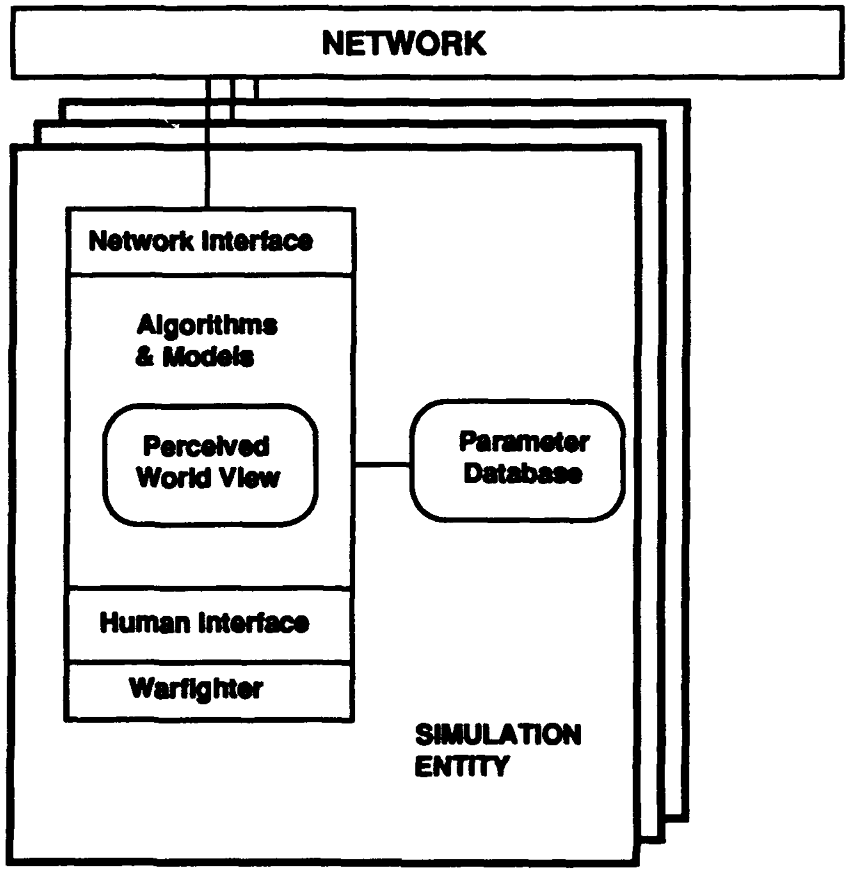


Figure 3.1-1: The basic elements of the DIS Reference Model are simulation entities and their interconnecting network.

DIS entities must conform to the basic DIS implementation principles:

- *The DIS system consists of networks of autonomous simulation entities interacting via networks, using a common environment database and compatible simulation models.*
- *Each entity maintains its own world view, based on its own simulation models, the common environment database, and state/event messages from relevant external entities.*
- *Messages are broadcast to signal significant simulation events and state changes. To reduce network bandwidth requirements, each entity uses remote entity approximation to project a continuous and coherent time/space view of relevant external entities.*

Each entity consists of closely coupled set of simulation components that:

- *Respond to operator control inputs.*

- *React to messages from other entities.*
- *Generate updated views of the surrounding environment and nearby entities for the operators at a rate sufficient to maintain the perception of continuity.*
- *Broadcast the occurrence of significant ownship events observable by other simulation entities.*

Since the focus is on human interaction, simulation time is generally real time.

Each entity incorporates a set of models and algorithms which define its simulation capabilites, level of fidelity, human interfaces, and interactions with other entities including the virtual environment. The parameters supplied to the models and algorithms define the common environment. Parameters include data defining terrain contour, texture, color, etc; data defining the shapes, textures, color, dynamics, fuel consumption, etc of vehicles; data defining atmospheric conditions, lighting, etc.; data defining simulation characteristics such as error thresholds, exercise plans, etc. This array of data is defined in considerable detail later in this discussion.

The messages transmit information about events as they occur (e.g. detonation of a projectile); the messages also update information about simulations of phenomena visible to other entities (e.g. position of moving vehicles).

The pattern of interaction between the inter-entity messages and the models and algorithms helps characterize DIS.

Computers simulate continuous processes by rapid computational iterations at rates sufficient to adequately approximate continuity. For traditional stand-alone simulators, various continuous processes constituting the overall simulation are linked by transfer of parameters at the iteration rate. This is generally a minor issue for centralized simulations, since shared memory or high speed buses provide easy data transfer. Distributed simulation can be performed using the same process, but long-distance data transfer is more expensive and data transfer latency increases by orders of magnitude to levels which can become directly perceptable to warfighter participants. Furthermore, a goal of DIS is interactive simulation of very large numbers of complex systems. The total message traffic and computational load must be carefully allocated if the goal is to be realized at an acceptable cost.

One means used in DIS to minimize inter-entity message load is based on identifying relative time-invariant parameters that describe continuous processes. For instance, vehicle position information is described by transmitting vehicle positions in a message containing current position and current vehicle velocity vector. If the vehicle is traveling at a constant velocity, its position can easily be computed as a function of time until the velocity changes, using the process known as remote entity approximation. Thus one message serves to define the position of the vehicle, even though the position is changing continuously. Higher order position algorithms can be used to describe constantly accelerating vehicles, further reducing position update requirements. (The term "remote entity approximation" is used rather than the more frequently used "dead reckoning", since the latter implies a first order (velocity only) projection. Remote entity approximation includes first order and higher order projection.)

The net effect of this process is to reduce inter-entity message traffic at the expense of local computational load. The process requires that all entities agree in advance on message syntax, on simulation algorithms (or message interpretation), and on relevant parameters required by the algorithms (such as position error threshold). Advance agreement is required because none of this information is incorporated in the message itself. Therefore the architecture must provide a means to define and establish these advance agreements.

Similar logic applies to event messages. Event messages are transmitted in condensed notation based on prior agreement between simulation entities. The event messages generally invoke pre-defined processes and depend on pre-stored parameters. For example, projectile hit information is transmitted using message information which defines where the projectile hit and what type projectile was used. The receiving entity determines its own damage by invoking a process, which is typically based on a Monte Carlo process using pre-stored damage tables.

Currently, the only continuous-process simulations defined by DIS messages describe position and orientation of vehicles. (Speech is also a continuous process defined by DIS messages, but speech communication between human participants is actual speech, not simulated speech. Standard speech data compression techniques in use throughout the telephone network are used to minimize bandwidth requirements.) Additional continuous process simulations will be required within DIS as other simulation dimensions are added to the battlefield. For the complex and rapidly changing interactions characteristic of electronic warfare, the amount of information required to fully characterize interactions becomes very large and the interaction rate becomes continuous or nearly continuous. To avoid requirements for nearly continuous message interchange and corresponding increases in network bandwidth, message definitions will be required which are based on relatively time-invariant descriptors. As with the current remote entity approximation position description process, these messages will rely on predefined algorithms and data within each entity.

The message-based interaction of DIS is a form of Object Based Design implementation. The objects are the various models which make up the simulation entities. Objects interact by passing messages, using previously agreed-upon data elements. In a rigorously implemented Object Oriented Design (OOD) which incorporates object inheritance, the message interpretation is guaranteed to be the same for all entities because only one set of models and data are shared by all entities. DIS, however, applies a more general form of OOD in which it is left to the implementer to ensure that all entities use matching models and data.

The implications of these DIS principles on time/space coherence are discussed in detail in Volume II, as are the implications and implementation feasibility of more rigorous application of OOD to the DIS architecture.

3.2 The DIS Cell

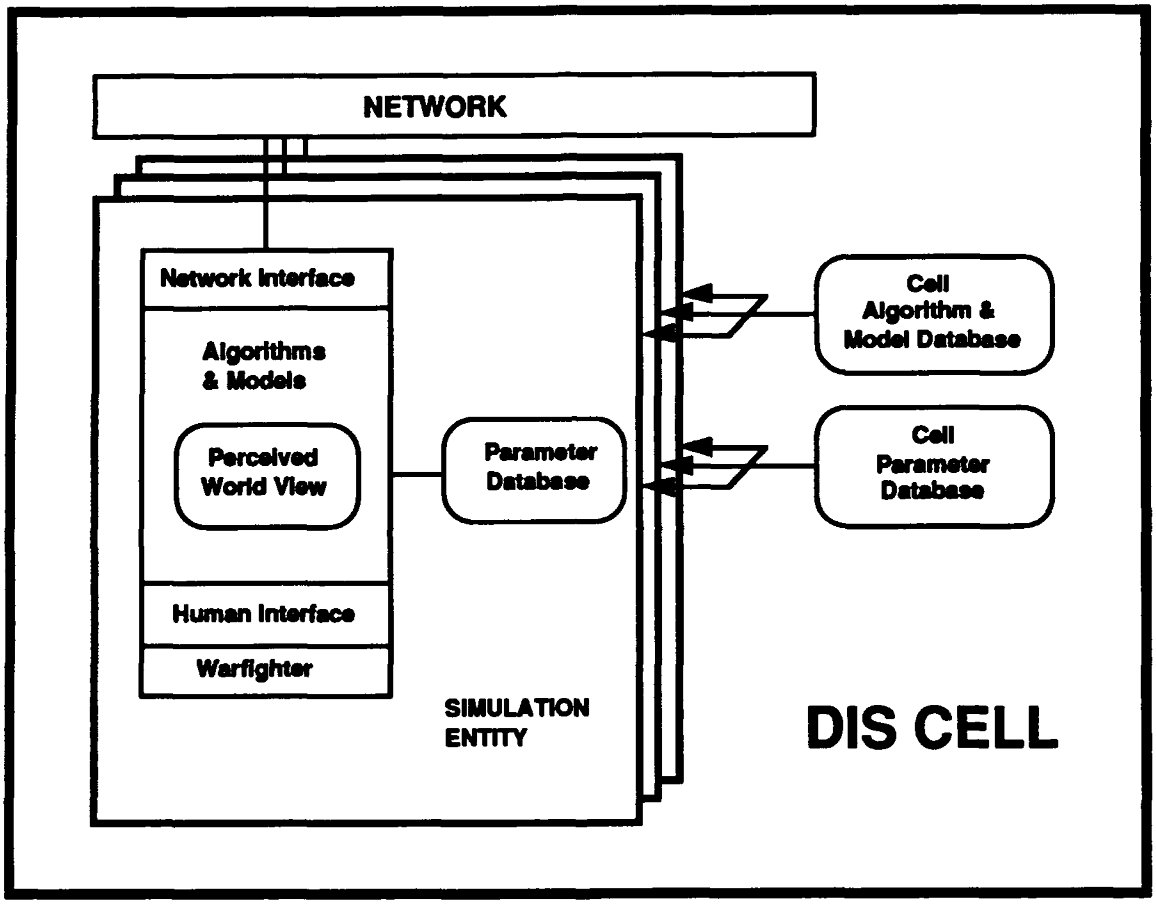


Figure 3.2-1: The DIS Cell is a collection of homogeneous simulation entities. All entities in a cell use fully compatible models and algorithms, share one set of data and parameters, and have unrestricted datagram interconnection via a network.

A DIS cell is a collection of homogeneous simulation entities connected by a network. To be considered homogeneous, a collection of simulation entities must all utilize the same parameter database, employ a fully compatible set of simulation algorithms and models, and have unrestricted broadcast of datagram messages from each entity to all other participating entities.

In simple terms, a cell is an interconnected set of simulators all using the same terrain database and compatible simulations; ie. the simulation models have been designed to work together. When the simulators require visual out-the-window view simulation, this usually means all simulators use the same display generation system design, since today each vendor uses unique display generation algorithms. For example, a set of interconnected SIMNET simulators using the same terrain database constitute a cell.

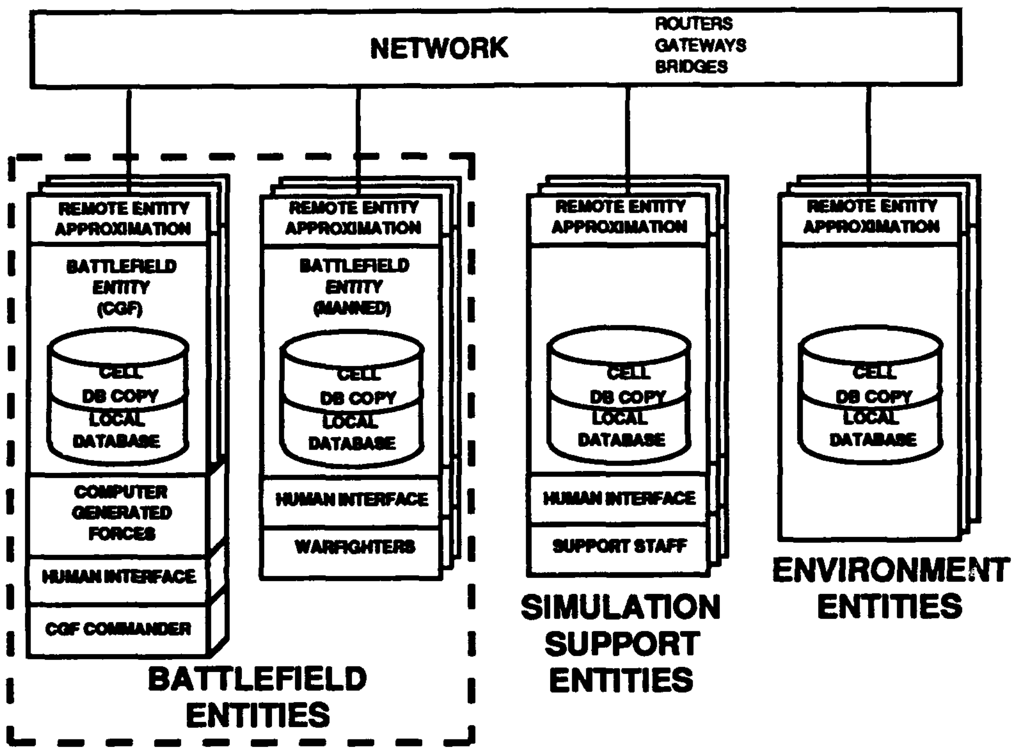


Figure 3.2-2: Three closely related entity types are defined for the DIS Reference Model.

Figure 3.2-2 illustrates a cell consisting of the three types of simulation entities used to create the DIS reference model. A fourth type of entity specific to Computer Generated Force (CGF) C3I is discussed in Volume II as part of a future standard DIS CGF architecture, but that architecture is not incorporated in the basic strawman DIS reference model. The C3I entity is postulated to distinguish the cognitive interaction between the elements of a distributed CGF from the physical interactions between battlefield entities.

Note that the entity illustration in Figure 3.2-2 is fundamentally the same as the prior illustration in Figure 3.2-1, but the entity internal components are shown at a higher level of detail. The "local database copy" is equivalent to the "perceived world view" in the prior illustration, and the "exercise database copy" equates to the "parameters database" of the first illustration.

Table 3.2-1 summarizes the primary characteristics of the three simulation entity types.

SIMULATION ENTITY TYPE	CHARACTERISTICS
Battlefield Entity	Corresponds to actual battlefield equipment or organization. Platform level battlefield entities include aircraft, ships, armor vehicles, dismounted infantry soldier, guided missile, command post, truck, ... Unit level entities, such as platoons, companies, etc. can also be defined. Incorporates direct soldier/machine interface which replicates soldier/machine interface with actual battlefield entity.
Simulation Support Entity	Simulation element which is incorporated to support or control the simulation, but has no equivalent on the actual battlefield. Examples include Plan View Display and "Magic Carpet" display.
Environment Entity	Corresponds to the components of the actual battlefield environment. Includes terrain (contour, surface, ..), atmosphere (haze, clouds, wind,...) /bathosphere, sun/moon lighting, and unmanned objects in the environment, such as trees, buildings, bridges, ... Has no direct soldier/machine interface, but takes responsibility for uncommanced obstacles, minefields, etc. that have been built or abandoned by battlefield entities.

Table 3.2-1: Three types of simulation entities constitute the basic simulation elements of the DIS Architecture.

Battlefield Entity: A DIS Implementation Principle requires that simulation entities correspond to actual battlefield entities. This principle is necessary to allow configuration of new simulation exercises corresponding to specific actual battlefields and engagements, and to allow development of simulated engagements incorporating new simulation entities corresponding to new battlefield entities. Both "battlefield entities" and "environment entities" correspond to actual battlefield objects; the difference between the two is primarily that battlefield entities have a direct human interface which replicates an actual battlefield soldier/machine interface, while environment entities have no human control (eg. cloud, ordinary landmine, bridge).

Note that the current Version 1.0 of the DIS PDU Standard (draft) considers all entities to be battlefield entities. The term "entity" as used in the draft PDU standard is the same as the term "battlefield entity" used in this discussion. In other words, the terms "environment entities" and "simulation support entities" are not yet in common use.

Battlefield entities include manned, automated, or semi-automated simulations. Manned battlefield simulation entities are operated by their warfighter crews, via man-machine interfaces that simulate the man-machine interface of the actual battlefield entity. The manned entities are the essential core of DIS, since DIS is focused on simulation of battlefield combat at the level of warfighter perception and interaction.

Automated battlefield entities include simulations of equipment and weapons which are unmanned on the actual battlefield (eg. target seeking missiles). Semi-automated entities are usually called Semi-Automated Forces (SAFOR) or Computer Generated Forces (CGF). CGF entities are battlefield entities controlled indirectly via a computer simulation of the crew operation and the higher level command structure which indirectly controls the crew, through a simulation of the battlefield command and control structure.

At the level of the reference model, manned or automated battlefield entities are equivalent. The objective of CGF systems are to generate simulations of actual battlefield entities that behave as much as possible like manned battlefield entities operating in a manned C2 environment, but with minimum manning. They fill out the battlefield, providing opposing forces and flanking/supporting forces, and they round out units when sufficient crews/simulators are not available or necessary to meet simulated engagement requirements. Therefore, from the perspective of the manned entities, CGF entities interact exactly like manned entities. They differ only from the perspective of the human interface; the manned entity human interface attempts to simulate the actual weapon and battlefield human interface, while the CGF human interface provides a simulation of the higher-level command interface. CGF simulates the C3I processes at levels from the individual crew commander up thru the command hierarchy. CGF must also provide computer simulation of crew control of the battlefield entity which responds to (interacts with) the simulated command hierarchy.

As discussed in Volume II of this document, there are strong reasons why a standard open architecture for DIS CGF systems should be created to allow modular development and modification of CGF capabilities. However, CGF standard architecture is not included in the strawman DIS architecture description which follows.

Battlefield Entity Level is discussed in Section 2.3, DIS Regime. The strawman architecture does not preclude incorporation of unit level entities. Incorporation of unit level entity interaction would require development of new PDU messages and simulation algorithms. It is also possible to incorporate unit level entities while restricting all battlefield interaction to the platform level. In that case, unit level entities must be de-aggregated to platform level when any interaction occurs.

Simulation Environment Entity: The environment includes the battlefield terrain, structural objects, ground cover, trafficability, weather, clouds (including smoke clouds), electromagnetic propagation characteristics, and nuclear/biological/chemical weapon effects, as well as ocean dynamics, acoustics, and sea state.

Often these environment entities are static throughout a simulation session, and are defined by an environmental database copied at each simulation entity. By providing a local copy of the data at each entity, rapid and continuous interaction with the simulation algorithm computation is facilitated.

The real environment changes dynamically, however. The changes may be the result of time-driven environmental simulations such as weather models, or caused by interactions with other simulation entities. Simulation of these changes requires dynamic changes to the environment database used by each simulation entity.

SIMULATION DRIVEN	INTERACTION WITH SIMULATION ENTITIES
Diurnal effects	Local smoke
Weather (fog, haze, rain, wind, ...)	Damage to structures
Trafficability changes caused by weather	Shell craters
Sea state	Trafficability changes caused by vehicle passage
Nuclear/biological/ chemical weapons effects (pre-scripted)	Combat engineering effects
	Nuclear/biological/ chemical weapons effects (interactive)

Table 3.2-2: Changes to the environment may be created by environment simulations or by interaction between the environment and other entities.

The environment simulations may simply be pre-defined changes to parameters stored in the environment database triggered at predetermined times, or may be the result of complex volumetric and/or electromagnetic models.

Dynamic changes must be consistent for all affected entities. If a bridge falls because of combat damage or because vehicles using the bridge exceed its load limit, the bridge must fall for all participants in an exercise; it can't remain standing for some battlefield entities but not others.

Consistency of dynamic environment can only be assured if a single environment entity controls the state of each dynamic component of the environment. For example, the total load on the bridge depends on distance between tanks as they cross, which in turn depends on positional accuracy of the REA. Unless there is a single controlling environment entity, different simulation entities could arrive at different conclusions as to the bridge collapse, since tank locations can vary between simulators (within the error bounds of remote entity approximation and inter-entity time latency differences) .

The elements of the environment are components of the real battlefield, and therefore the environment entity is very much like the other battlefield simulation entities. Like all other entities, environment entities must maintain a local view of all other relevant entities in order to determine own state. It differs in certain ways, however. It lacks a human/machine interface, and it sometimes interacts with other entities by distributing changes to the environment database (ie. dynamic terrain; see Volume II for further discussion). These differences lead to the decision to treat the environment as a special entity class in the DIS architecture. Definition of the environment entity also conforms to the object-oriented reference model decomposition, with its corresponding organizational benefits.

Simulation Support Entity: The reference model also includes simulation support entities. By definition, simulation support entities are simulation modules that have no direct equivalent on the battlefield but are required to support the simulation. Examples include:

- Devices which support utilization of the simulation, such as After Action Review facilities, plan view displays, and phantom vehicles (like the SIMNET "stealth").
- Devices which support operation of the simulation, such as control consoles.

Note that Simulation Support Entities have a special class of information and modeling privileges. Simulation Support Entities are allowed access to information that would not be available to actual battlefield entities and are not restricted to adherence to physical laws for maneuver, line of sight, etc. Likewise, simulation support entities are restricted from interacting with other entities during free play exercises. They interact only for exercise control purposes, such as initialization or for exercise controller intervention in an exercise.

These extended privileges are reflected in the common names given to the SIMNET version of the Simulation Support Entities. The "Magic Carpet" is an invisible vehicle (or Stealth, but this one is totally invisible since it has no graphic representation on the simulated battlefield), able to move at any velocity and acceleration. Support staff can use the Magic Carpet to view any part of the simulated battlefield as the battle unfolds. Likewise, the Plan View Display, which presents a real-time map view of the battlefield, including vehicles and events, is known as the "god's eye view".

As an aid to discussion, cells are divided into two categories:

DIS Standard Cell: A Standard DIS Cell is simply a cell which conforms to DIS Standards. As will be discussed below, the DIS Standards define the messages PDUs which flow between entities, the models and algorithms incorporated in the entities, the parameter databases associated with the models and algorithms, and communication protocols used to carry PDUs, and certain simulation control functions and interfaces.

Non-standard Cell: Non-standard cells are collections of simulation entities of any type that do not meet the DIS Standard Cell criteria, either because the cell does not fit the DIS Cell model as described above, or because the DIS standards were not applied to the cell. The current SIMNET cells are non-standard because current SIMNET sites do not conform to the only published DIS Standard (the draft SIMNET PDU Standard). Other examples of non-standard cells include all existing high-fidelity simulators, actual equipment instrumented for operation on all existing tactical engagement ranges, and all existing high order model simulations.

Note that it is possible to develop DIS Standard cells for all of these types of simulations, but to date the focus of the DIS community has been on simulation networks modeled on the SIMNET prototype, and the initial DIS standards are optimized for similar systems. Other types of simulations will likely require extensions to the standards. For instance the draft PDU standards tacitly assume availability of relatively unconstrained inter-entity message bandwidth with high communication reliability. The radio communications used between simulation entities on instrumented ranges is less reliable and far more bandwidth constrained than the LAN/LHN networks used by the current DIS simulators. Extensions to the standard (or creation of compatible versions of the standard) optimized for the special requirements of range simulations may be warranted.

Heterogeneous Cells: The DIS Architecture supports the DIS goals of multiple simultaneous exercises and seamless interoperation of heterogeneous simulators by supporting interoperation of multiple cells.

Multiple simultaneous exercises are supported by partitioning the overall array of DIS entities into multiple independent virtual networks. If all of the simulation entities on given virtual network are using the same models, the same data, and are interacting via broadcast datagrams, they constitute a cell. Thus the current array of SIMNET assets can be configured to be one large cell or multiple independent cells, but each cell must use the same database.

The most challenging and important goal of the DIS Architecture is provision to support interoperation of heterogeneous cells. There are a multitude of reasonable ways in which cells can differ yet interoperate in a valid way. Of course, the term "valid" always implies "valid for the purpose intended". Validity can not be determined independent of user needs and requirements.

Interoperation requires at least some degree of correlation between the interoperating cells, but the degree of correlation required is entirely dependent on the application. The following list provides a number of examples of how cells can differ yet interoperate.

Differing Visual Rendering Capability and Associated Differing Terrain Database Degree of Detail: Computer Image Generation (CIG) and associated Display Systems have received a great deal of attention within the DIS community. It is clearly not practical to require all CIGs to use just one set of display system algorithms and parameters, or to standardize on one set of algorithms implemented with a few standard sets of parameters. Visual system cost is very often a major part of the cost of a simulator, and various visual systems are available which offer significant cost/performance advantages for certain classes of functions.

Cells with different CIGs can interoperate successfully in many ways. Sometimes the CIG differences are not a fidelity limitation at all. For example, a weapon system may actually have different visual capabilities for different operator positions, such that realism requires correspondingly different simulated visual capabilities at each position. However, it will sometimes be necessary to tolerate visual system capability differences between essentially equivalent participants. Often the degree of difference will not be critical for the exercise. In other cases, the exercise can be planned such that the participants in one cell use a different part of the simulated battlefield and never make visual contact with the players in the other cell: they may represent flanking forces, or they may be opposing forces engaged using only long range weapons. Alternatively, CIG rendering parameters can be adjusted to help equalize visual performance, even if doing so requires reducing certain capabilities of the more sophisticated CIGs, or the visual model database can be adjusted to compensate for different rendering capability (eg. colors can be adjusted to make targets easier to detect at long ranges on lower resolution CIGs). In all such cases, the validity of the interoperation can only be evaluated in terms of the exercise objective.

Differing Time/Space Accuracy Requirements: Cells focused on high speed/high acceleration entities such as aircraft may require far tighter time/space accuracy constraints, compared to cells populated by slower entities such as armor or infanty. The cells may therefore use differing REA algorithms and error bounds, resulting in greatly different PDU rates for the different entity types.

Differing Inter-entity Message Constraints: The current DIS PDU standard tacitly assumes availability of relatively inexpensive message bandwidth and relatively high communication reliability. The constraints may be different for some types of DIS cells, such as actual equipment operating on instrumented ranges where radio-based message bandwidth and reliability is low compared to LAN/LHN nets. Such cells may require use of PDU messages optimized for the available communication channels, requiring translation to interoperate with simulator cells.

Differing Degrees of Model Fidelity: Visual scene fidelity has already been discussed. Other elements of the simulation also differ greatly in fidelity requirements between user domains. In particular, Electronic Warfare fidelity requirements often differ greatly between ground combat, naval surface warfare, air-to-air combat, and air defense. Users in each domain require modeling accuracy with emphasis on different aspects of the EW environment. Fidelity requirements differ between user communities on various other aspects of the overall simulation.

Ultimately, the goal of DIS is to support valid interoperation without imposing excessive costs. That goal requires that the DIS architecture support interoperation of cells that have been optimized to support specific user focuses, without requiring all users to agree to a single fidelity and simulation solution.

3.3 The Intercell Tier of the Network

The DIS architecture reference model interconnects heterogeneous cells via a second network tier. Each standard cell contains a inter-entity network which is typically (but not necessarily) a LAN. The LAN in each cell connects to the intercell network, which is typically a Long Haul Network, via Cell Interface Units (CIU), as illustrated by Figure 3.3-1. Non-standard cells may or may not contain an internal network, but the simulation entities within the cell connect to the intercell network via Cell Adapter Units (CAU). Each is explained in greater detail below.

The intercell network must establish virtual data circuits between the cells, and provide data rates, end-to-end latency, and reliability within performance bounds specified for a particular multicell exercise. Commercial service protocols and standards are emerging which provide the required service levels, as discussed in detail in Section 4 of this document.

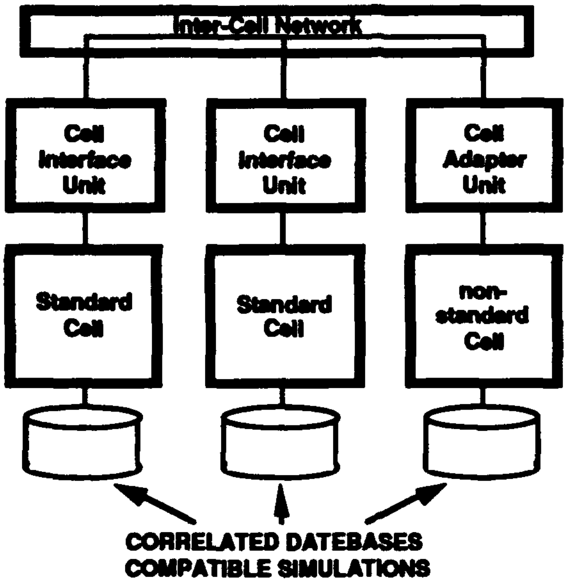


Figure 3.3-1: A multi-cell exercise consisting of two Standard Cells and a non-standard cell.

Cell Interface Unit (CIU): The CIU is included to help cope with interoperation of cells and to provide a means to control network traffic load on the upper tier of the network. CIUs can perform either or both functions. For example, two networks of entities located at distant sites may be identical and therefore capable of operating as one homogeneous cell, but CIUs may be inserted between them in order to reduce long haul network traffic. In other cases, two substantially different heterogeneous cells may be interconnected via CIUs, with the CIU performing message translation to achieve adequate interoperability.

A number of classes of CIU functions are envisioned:

Message Filtering: Message filtering is defined as blocking messages from the upper network tier based on message contents. Note that this function differs from message routing, which is a function performed by the network. Routing creates virtual networks from general purpose networks, based on message addressing schemes. Filtering blocks messages by analyzing message contents. As a simple example, two cells may be organized so that tactical radio traffic between the cells is eliminated or restricted to a subset of the available (simulated) radio nets within each cell. In that case, radio message data for all or most simulated nets would be blocked from the second network tier by the CIU. Many other types of filtering can be envisioned, including elimination of messages that are irrelevant to other cells due to fidelity differences.

Filtering serves two purposes: it reduces inter-cell message traffic and, by doing so, it reduces the number of messages that have to be sorted and analyzed for relevance at the input to each entity. For very large scale exercises, this processing load could easily overload simulators.

Aggregation/Deaggregation: Large scale exercises may be organized to incorporate multiple homogeneous and/or heterogeneous cells, with separate operational units occupying separate parts of the virtual battlefield. At least for some purposes, it may be possible for the distant units to interact only at higher levels of aggregation (eg. platoon, flight). In such cases, only information about the aggregate status of the unit need be transmitted on the second tier of the network The CIU at the transmitting end would determine the net status of the unit from the individual entity status within the cell (aggregation).. The receiving CIU would perform the inverse function, regenerating an approximation of each individual entity status if necessary (deaggregation).

Similarly, the sending cell might be contain a high-order simulation which models only unit (multi-entity) behavior. In that case, the high-order model and the DIS cell could interact via the aggregation/deaggregation function of a CIU.

Compression/Decompression: The DIS PDU format offers opportunity for content-based data compression techniques.

Encryption: As discussed in Section 5, the cell boundary provides a convenient security boundary. The CIU can encode/decode messages prior to handover to the open network.

Translation: The CIU also allows for insertion of an "intelligent intermediary" between heterogeneous cells that can interpret message from one cell and translate them to be more meaningful in the context of the models and database of the other cell. For example, for two cells with greatly differing Electronic Warfare fidelity may use incompatible RF modeling algorithms. One may rely on complex atmospheric propagation models, while another may use a simple line-of-sight model or no attenuation model. The CIU could use the complex model to make detection decisions for the low fidelity cell, generating messages as needed to simulate appropriate detection behaviors in the low fidelity cell (eg. to control radar warning receivers). The feasibility of such translation between DIS cells remains to be demonstrated, however.

Cell Adapter Unit (CAU): CAUs translate the information in a non-DIS cell to DIS messages, plus they perform all of the functions performed by CIUs. The non-DIS cell could be entire simulator network (eg. SIMNET or SOF ATS), a single simulator (eg. a high-fidelity weapon system trainer), a high-order model, an actual battlefield entity (eg. a Maneuver Control System fieldstation), or an instrumented range that does not use a DIS-compliant internal message format.

For example, a typical stand-alone simulator that performs complete state calculations at a rapid iteration rate has no need for a remote entity approximation capability (except for frame by frame interpolation). The entire state of the simulator and its threat environment might be defined by a memory-resident state table that is updated each computational frame. To connect such an entity to DIS, the CAU would be required to perform the remote entity approximation and event threshold detection to generate outgoing messages, and to update the simulator internal state table as message arrive.

In a similar fashion, a CAU can translate the entity state vectors maintained by the central range control computer for each actual equipment entity on an instrumented range into DIS message format. Likewise, the CAU can translate DIS messages from the network into range entity control messages.

Obviously the functions of a CAU cannot be standardized, but to some degree they can be generalized, since most existing frame based simulators are somewhat similar internally, as are most existing engagement ranges.

3.4 The DIS Reference Model

The DIS Reference Model shown in Figure 3.4-1 summarizes the entity, cell, and intercell layers of the model discussed in Sections 3.1, 3.2, and 3.3. The elements of the model have been addressed by the preceding discussion; this figure summarizes the elements in a single reference model. The figure also introduces a naming convention for various versions of the database as applied to a multicell configuration. The database structure is discussed in Section 3.6 below.

For clarity, the figure shows both manned and semi-automated battlefield entities.

The figure adds some internal details to the CIU/CAU, indicating that the CIU & CAU may maintain a perceived world view using the database and message traffic. Maintenance of such of world view may not always be required, depending on the functions performed for a given exercise. The CAU interface to the non-standard cell is by definition specific to the particular cell.

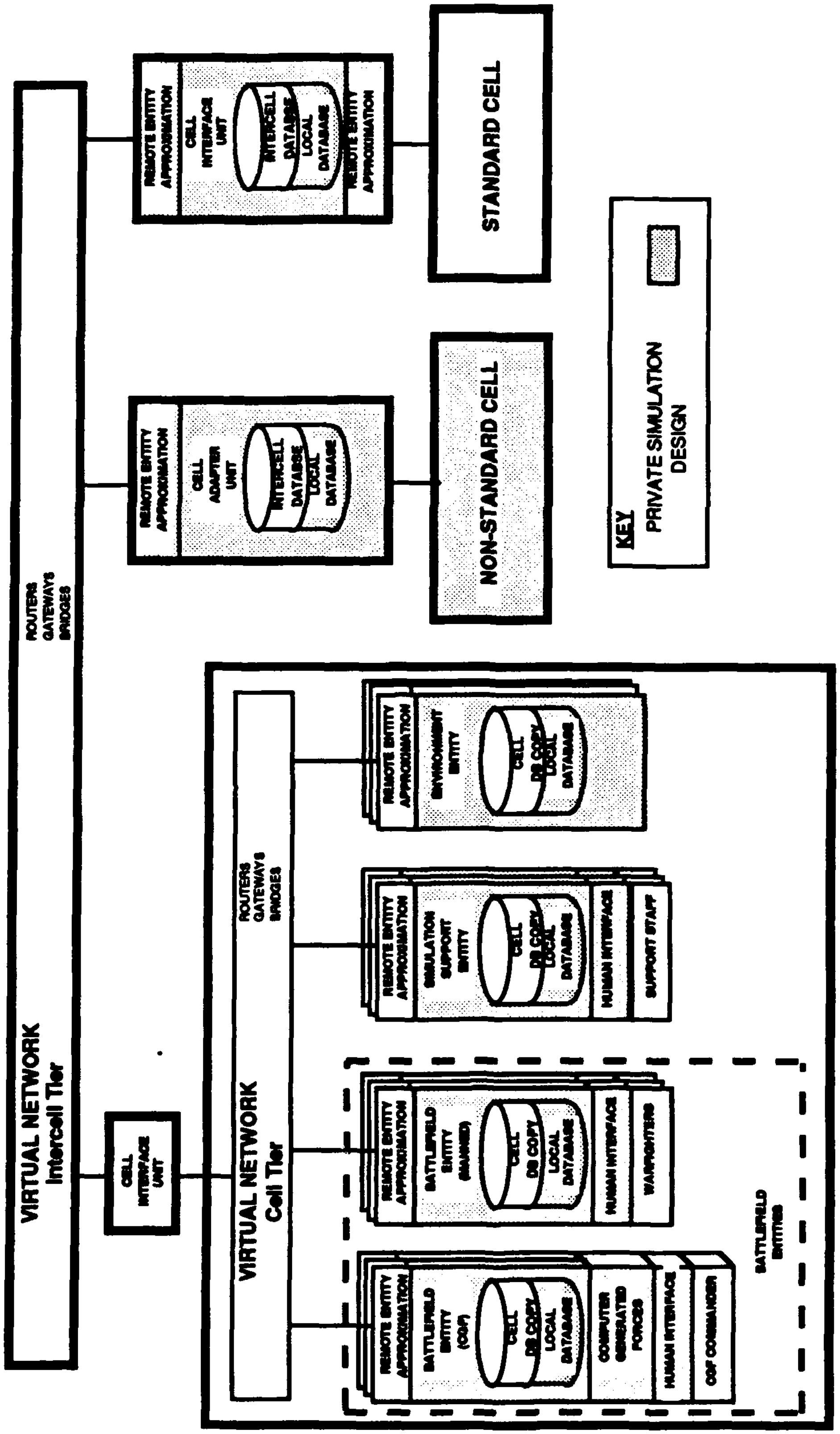


Figure 3.4-1: DIS Reference Model and Standards

3.5 DIS PDU Message Standard

Messages between all of the entities which comprise the reference model are consistent with DIS PDU Message Standard currently being defined by the DIS community under joint PM TRADE and DARPA sponsorship ("Standard for Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation", draft version 1.0). The supporting communication architecture can be defined within current and emerging industry networking standards, as discussed in Section 5.2.

However, the draft standard focuses almost exclusively on messages between battlefield entities. As noted early, the term "entity" as used in the draft standard is equivalent to the the term "battlefield entity" in the reference model. One reason to distinguish between the entity types is that the three types of entities interact in different ways. For example:

- Simulation Support Entities (SSEs) will issue control messages such as the Activate and Deactivate PDUs of DIS 1.0. SSEs will exchange control messages, such as the Plan View Display, Stealth, and Data Logger protocols of SIMNET.
- Battlefield Entities (BEs) will exchange combat events including the messages defined by the DIS PDU Standard.
- Simulation Environment Entities (SEEs) will issue environmental change messages to Battlefield Entities.

Further, CIUs and CAUs may have to exchange messages in order to optimize the flow of information between Cells across the Intercell Network.

Figure 3.5-1 presents a notional family of DIS Protocols.

The members of this protocol family can be segregated by assigning unique protocol numbers to each member. Receiving entities can find this unique protocol number in a fixed, well-known place in each packet (the "header"). The protocol number enables entities to easily and efficiently accept or reject packets based only on the unique number, thus increasing the number of entities that can be supported by the system.

This segregation of the DIS protocols also enables the use of "concurrent engineering" techniques to minimize development time, effort, and cost. These techniques include concurrent prototyping and parallel development for each of the protocols.

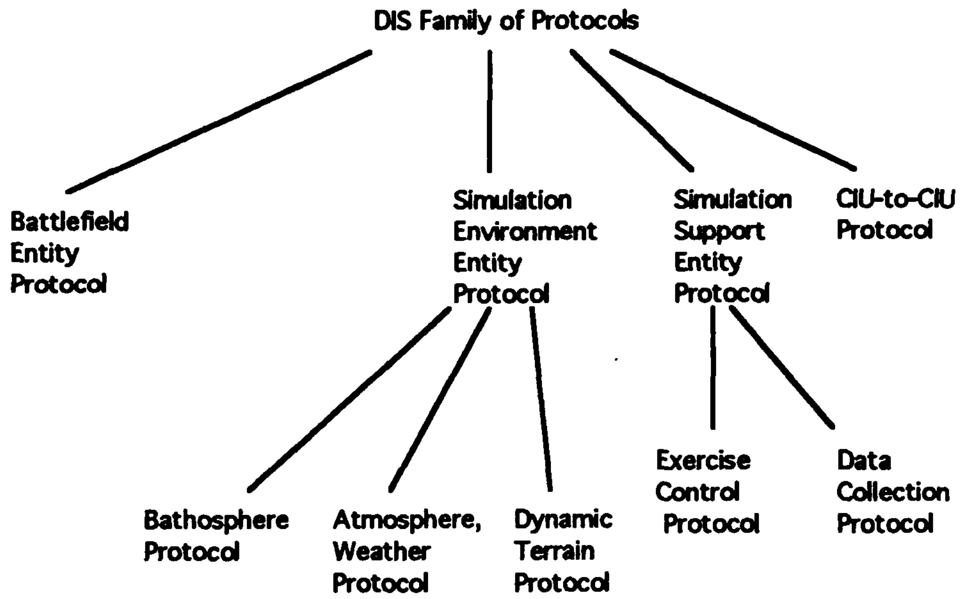


Figure 3.5-1: DIS Protocols

3.6 DIS Standard Cell Database

In summary of much of the preceding, interoperation of cells requires:

- *Use of a compatible set of models and algorithms.*
- *Use of correlated databases.*
- *State exchange between entities using a common message protocol.*
- *Use of compatible communication networks.*
- *Use of some common exercise control process.*
- *That users have ready access to the information needed to assess the validity of interoperation for a specific application.*

Compatibility and correlation are assured if the models, algorithms and associated parameters, and databases are identical for all entities. The challenge of DIS is interoperation between cells when any of these factors differ. The first objective of DIS is standardization on as many factors as practical, thereby avoiding barriers to interoperation. Where differences in user focus, fidelity, and capacity force simulation differences, the goal becomes elimination or reduction of the barriers to valid interoperation.

It is impractical to require that all simulators utilize just one type of computer image generator. It is equally impractical to require that all entities use one atmospheric model, electromagnetic propagation model, radar and IR model, etc.

Different applications of simulation have different needs, resulting in requirements for different degrees of fidelity and different types of outputs. The DIS architecture is based on the premise that the architecture must maximize interoperability while allowing for a variety of such simulator and simulation implementations. In fact, new models are continuously being developed to explore new dimensions of problems, to improve simulation fidelity and efficiency, and to model new battlefield systems.

When it becomes necessary to utilize multiple databases & models, the objective of the the DIS architecture is achievement of interoperability with a reasonable amount of effort and time. In general, the overall database and model families used for a multicell application will be identical in many regards, but will differ in limited areas. However, as DIS expands, the range of differences will also expand and the degree of heterogeneity will increase.

To date, the DIS community has focused on development of a message PDU standard as the primary means to achieve the interoperability goals of DIS.

This strawman architecture document proposes creation of DIS Cell Database Standard as an additional means to promote the DIS interoperation goals. The new standard is intended to provide a standardized means to:

- *Define the models and algorithms used by the entities within a cell.*
- *Specify the data requirements of the models and algorithms.*
- *Define a data exchange format for the parameters and data required for application of the cell.*
- *Define the network communication standards for both network tiers for a specific cell application, and provide the required network management information.*
- *Provide the information needed by users of the overall simulation to analyze fidelity, analyze degree of correlation, and determine validity of exercises involving one or multiple cells.*
- *Provide the information needed by users to develop means to improve interoperability of multicell exercises, including the information needed to define and develop CIU/CAU functions.*
- *Provide the initial condition data required by cells for a specific cell application.*
- *Provide the information required to reduce inter-cell network message traffic, when network cost, network capacity, or simulation node overload becomes a limiting factor for a multicell exercise.*

- *Define the data and an associated standard data exchange format for the information required to coordinate and manage multicell exercises.*

Standardization of the DIS Cell database represents a new major effort, parallel to the current DIS PDU standardization effort. The new standard could well incorporate many of the DoD simulator standardization efforts, including the Air Force Project 2851 efforts to define DoD-wide simulated terrain and graphic model databases, and possibly the Navy Universal Threat Simulation System, but the new standard must deal with a number of new areas not currently included in the existing efforts.

The postulated DIS Cell Database is divided into three major components:

- *SIMWORLD Database is a collection of specifications defining the simulation models and algorithms used by a collection of simulation entities. These specifications define the data and parameters required by the models and algorithms, which are supplied by the BATTLEFIELD and SESSION databases.*

- *BATTLEFIELD Database, defining the specific data and parameters to be used by a collection of entities for a series of exercises.*

- *SESSION Database, defining the initial conditions for a specific exercise, and the network topology required to support the exercise.*

SIMWORLD	BATTLEFIELD	SESSION
Defines simulator characteristics/ fidelity/algorithms	Defines model data and parameters (graphic representations, geography...)	Defines initial conditions and network topology/connectivity
examples:	examples:	examples:
Terrain model	Terrain location, geography, features	Dynamic terrain initial conditions
Vehicle models	Vehicle graphic representations	Vehicle Initial conditions (fuel, ammo)
Weather model	Weather parameters	Initial weather condition, time of day
Remote Entity Approximation (REA) models	REA error threshold parameters	Initial entity positions
Message types (PDUs) allowed	Communication standard profile	Entity and network topology

Table 3.6-1: The DIS Cell Database is divided into SIMWORLD, BATTLEFIELD, and SESSION components.

The SIMWORLD fully characterizes the entities in a cell, except for the exercise specific data load. If entities are members of the same SIMWORLD, they can function in the same cell if they use the same BATTLEFIELD data load. The SESSION defines a particular simulation exercise, including the BATTLEFIELD to be used for the exercise.

The segmentation of the Cell Database is based on the premise that a small number of SIMWORLD databases can be defined that meet the needs of most DIS users. Likewise, within each such SIMWORLD, a few standard BATTLEFIELD databases will meet most needs. Most users will consistently work within one SIMWORLD, using a small number of BATTLEFIELDs. SESSION databases are likely to change frequently, but the amount of information in the SESSION database is not large. The SESSION database includes all of the information required to establish and initialize an exercise, once the SIMWORLD and BATTLEFIELD have been agreed upon. The SESSION database supports easy establishment of a virtual network for the exercise and the information needed to support exercise control.

When an exercise requires interoperating of cells across multiple SIMWORLDs, the Cell (and intercell) databases provide a single source of virtually all of the information required for potential users to develop correlated versions of the BATTLEFIELD, determine the simulation validity of the combined suite of assets for the intended purpose, and coordinate the exercise. These databases also provide the information necessary to determine the need and functional requirements for Cell Interface Units.

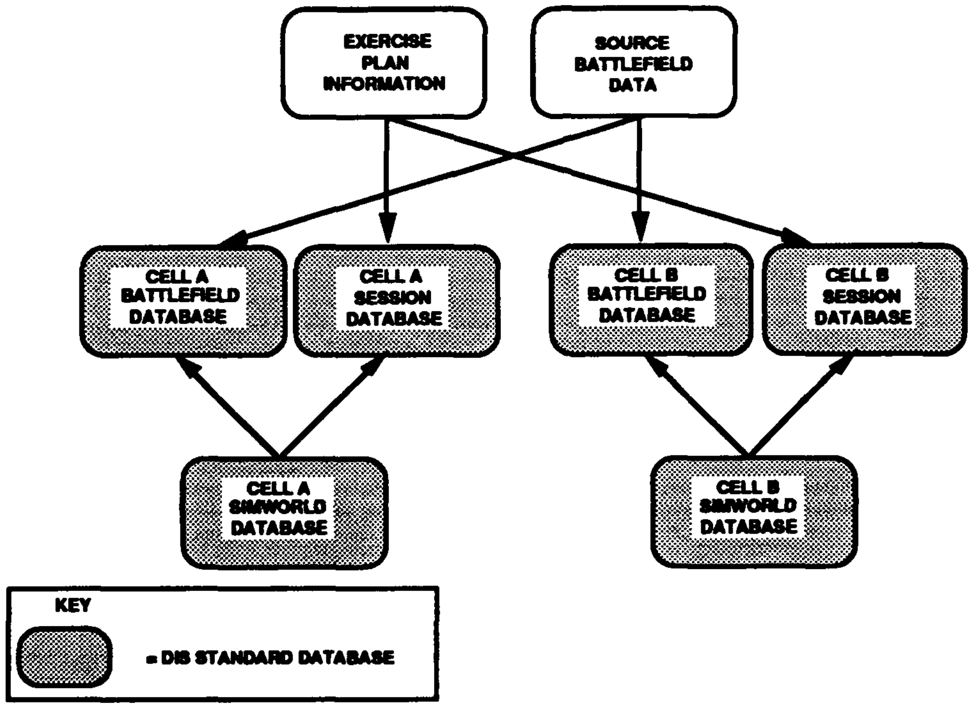


Figure 3.6-1: The SIMWORLD defines data requirements and exercise coordination requirements for multicell exercises in terms of the particular requirements for each cell as derived from the overall exercise plan.

Figure 3.6-1 conceptually shows the relationship between the databases for a two cell exercise. Assuming that the entities in each cell are members of different SIMWORLDs, the SIMWORLD defines the exercise plan data requirements for each cell; the exercise data is provided by the SESSION data base. The SIMWORLD also defines the versions of the source battlefield information required to define the BATTLEFIELD database for each cell.

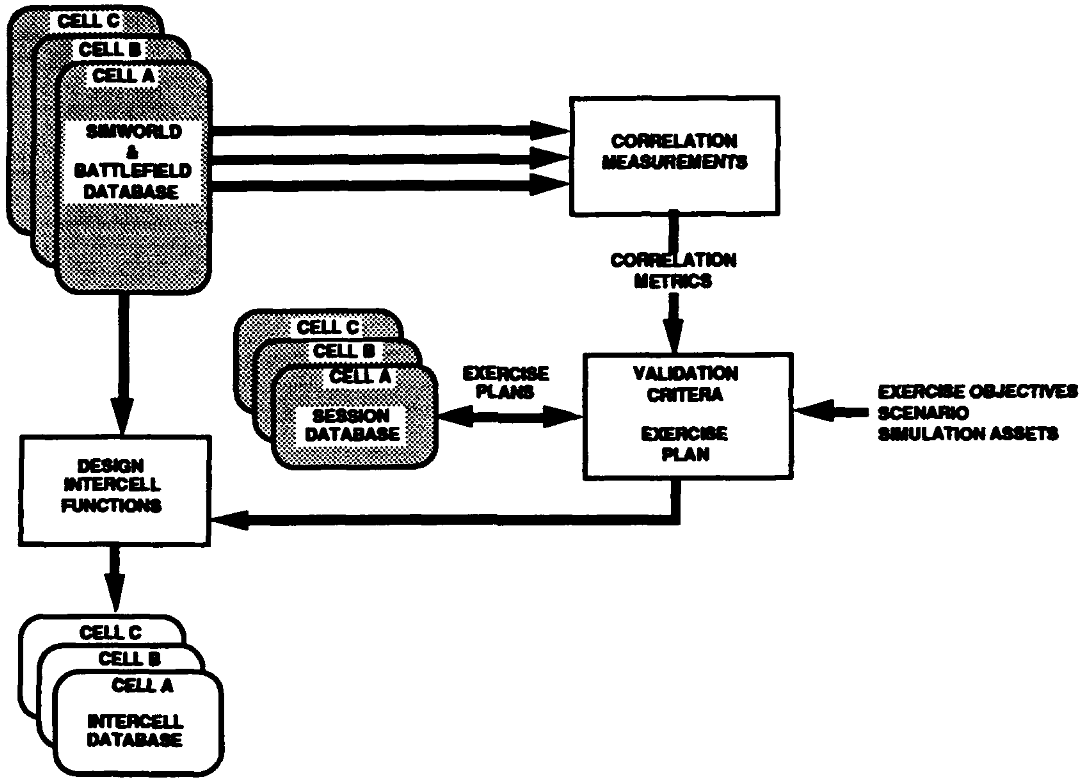


Figure 3.6-2 The components of the Cell Database provide the information required to plan multicell exercises, including the information needed to evaluate validity and loading. If necessary, the information can be used to modify cell interactions via the CIU to improve validity and to adjust network loading.

As illustrated by Figure 3.6-2, the combination of the SIMWORLD and BATTLEFIELD provide all of the information needed to generate measures of correlation, assess the validity of the exercise, determine the need for fidelity adjustments, and evaluate the feasibility of implementing such adjustments. Of course, the degree of correlation may be determined to be acceptable without adjustments. Requirements for CIU functions can be determined, and exercise plans can be defined based on correlation, exercise objectives, practical considerations such as LHN costs, security requirements, etc. Once the CIU functions have been determined, any databases required by the CIU can be generated from the SIMWORLD, BATTLEFIELD, and SESSION. The term "INTERCELL Database" is introduced here as the class name for CIU and CAU databases. INTERCELL databases cannot be generalized, however, since they must be designed on a case by case basis. The INTERCELL database is therefore not considered part of the DIS Database Standard, but it is derived from the SESSION, BATTLEFIELD, and SIMWORLD components which comprise the Cell Databases of the various cells in an exercise.

Note that the database standardization does not imply a requirement that all DIS entities incorporate an automated capability to adapt to any SIMWORLD by automatically interpreting the SIMWORLD database. It may be possible to build in some degree of automated adaptability, but it is sufficient if an entity can be fully characterized by the SIMWORLD and BATTLEFIELD information. Most simulators will be designed to function in one particular SIMWORLD, and will not be adaptable at the SIMWORLD level. The simulator will be fully characterized by its SIMWORLD membership.

Certain DIS messages support initialization and/or transmission of the common database over the network within the DIS protocol, except for dynamic components of the database. The reference model does not require that these messages be defined. It is expected that media or other standard data nets would be used to intialize data bases prior to exercise initiation. The SESSION database includes the information needed to establish the virtual network for an exercise. At least for multi-cell exercises, the network component must be provided to all participating cells prior to establishment of the session network.

The Cell Database provides a means to establish and maintain configuration management for DIS entities and events. The SIMWORLD database defines the functional capabilities of entities composing a cell. In general, entities will be designed to function in one particular SIMWORLD, and the SIMWORLD definition specifies virtually all of the public aspects of an entity. SIMWORLD definition therefore provides a means to establish controls on proliferation of entity types. In other words, policy can require that all simulators of a given class must conform to a specific SIMWORLD.

BATTLEFIELDs define specific instances of SIMWORLDs. The specific case may be derived from a source database representing a real world battlefield, or may be a hypothetical battlefield. If it is hypothetical, it may well serve as the source database for correlated BATTLEFIELDs for related SIMWORLDs. In either case, the BATTLEFIELD can be validated and configuration managed for use over multiple applications.

SESSION focuses on planning, coordination, and definition of specific exercises, and does not require the same degree of configuration control as the more permanent parts of the database. As large scale DIS exercises involving multiple cells become regular events, it may prove desirable to develop multi-user/multiple access software tools to support generation of standard SESSION databases.

3.7 Strawman DIS Architecture

Figure 3.7-1 shows the DIS reference model and the relationship of the DIS PDU Message and DIS Database Standard, resulting in the strawman DIS Architecture. Figure 3.7-2 shows the hierarchical relationships within the architecture.

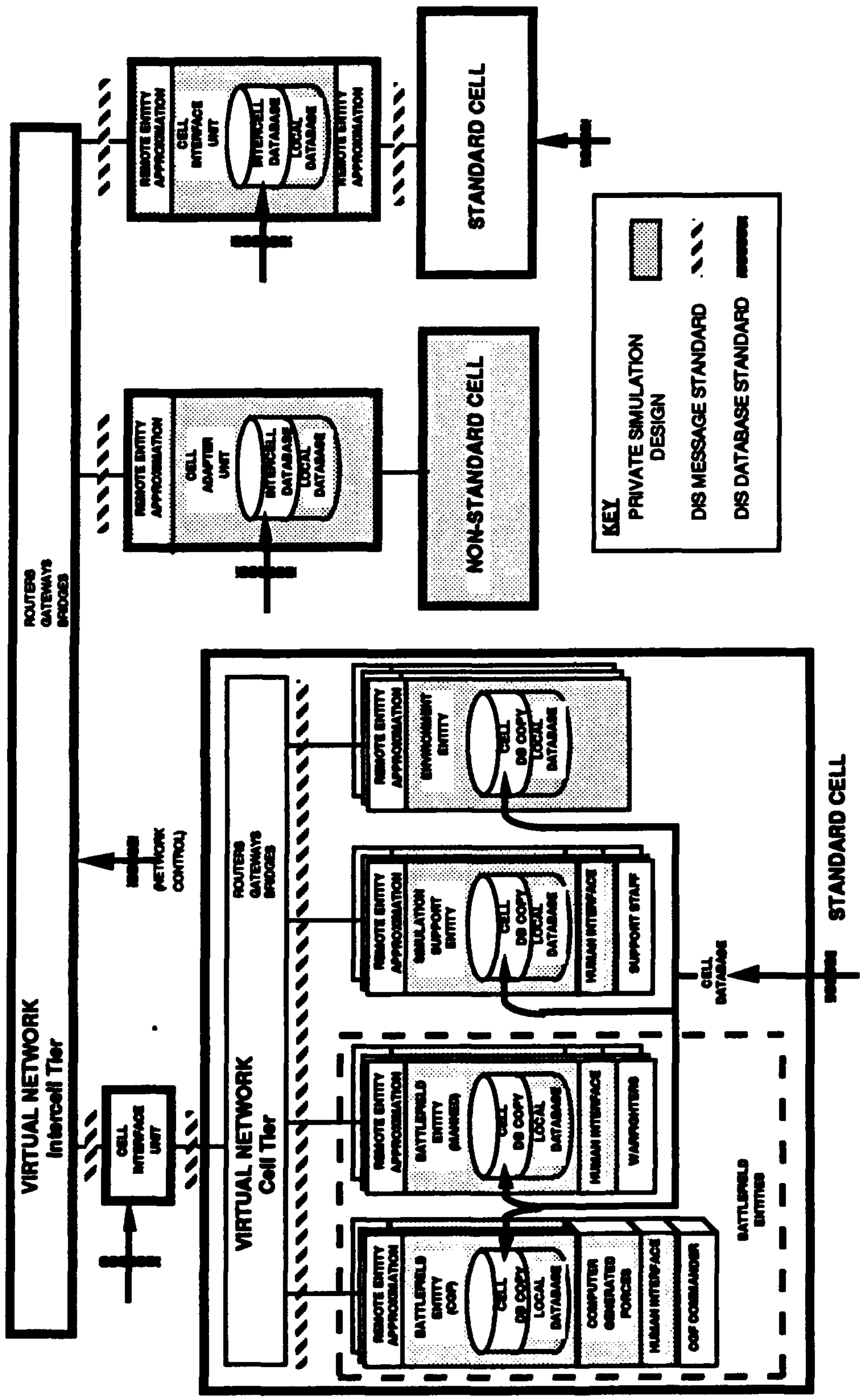


Figure 3.7-1: DIS Strawman Architecture

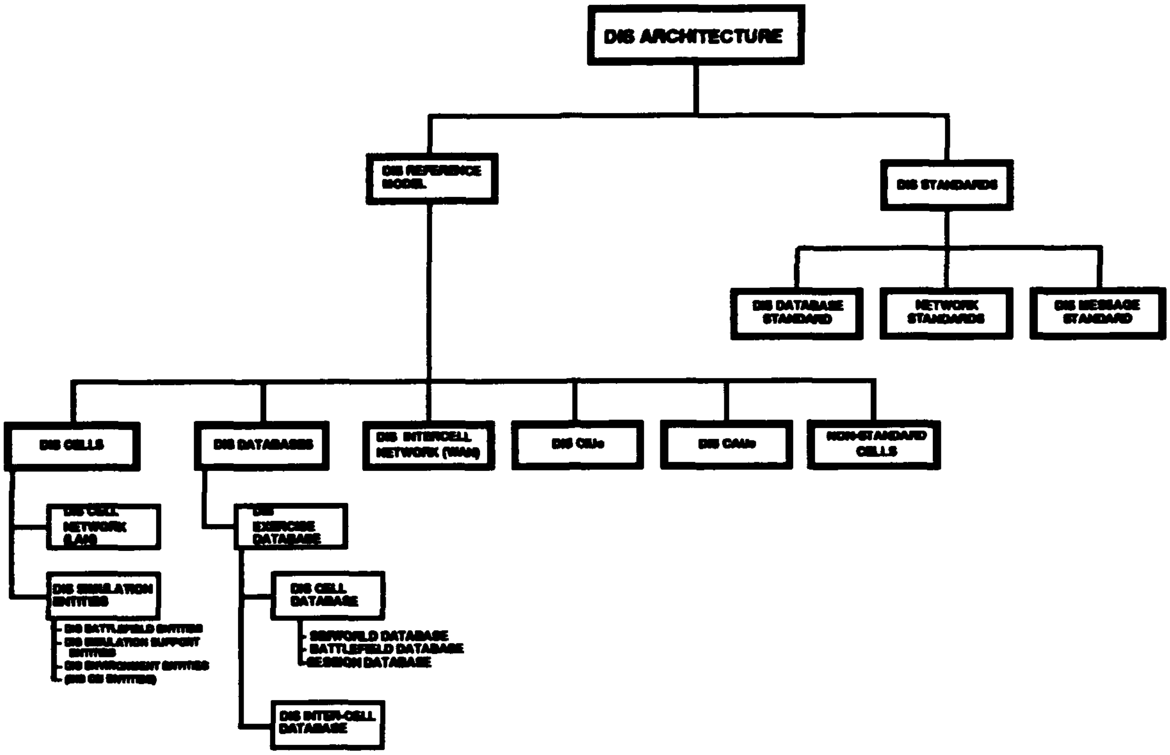


Figure 3.7-2: The hierarchy of the strawman DIS Architecture.

The remaining sections of this volume provide more detailed and more rigorous information on the DIS Architecture and Standards, and briefly discuss security, policy, and VV&A considerations. Volume II presents a technical basis and rationale leading to the strawman architecture.

3.8 Examples

Figure 3.8-1 is a simple illustration of some of the concepts of the preceding discussion. In the example, the entities at site 1 have been partitioned into two cells, W and X. The entities at the site might all be members of the same SIMWORLD, but the two cells are using different BATTLEFIELD and/or SESSION databases to support exercises A and B. Cell W is partcipating in a exercise A with cell Y at site 2, via a long haul network. Because cell W and Y are members of different SIMWORLDs, each cell in exercise A uses its own correlated version of the cell database. In this illustration, Cell Interface Units are being used to facilitate the two-cell exercise, but CIUs might not be required if no filtering is necessary and the cells are sufficiently well correlated without inter-cell translation.

The entities at site 3 and 4 are all members of a single SIMWORLD, and are sharing the same BATTLEFIELD and SESSION databases. No network filtering is being used. All of the entities at sites 3 and 4 are therefore members of the same cell, even though they are distributed accross two sites.

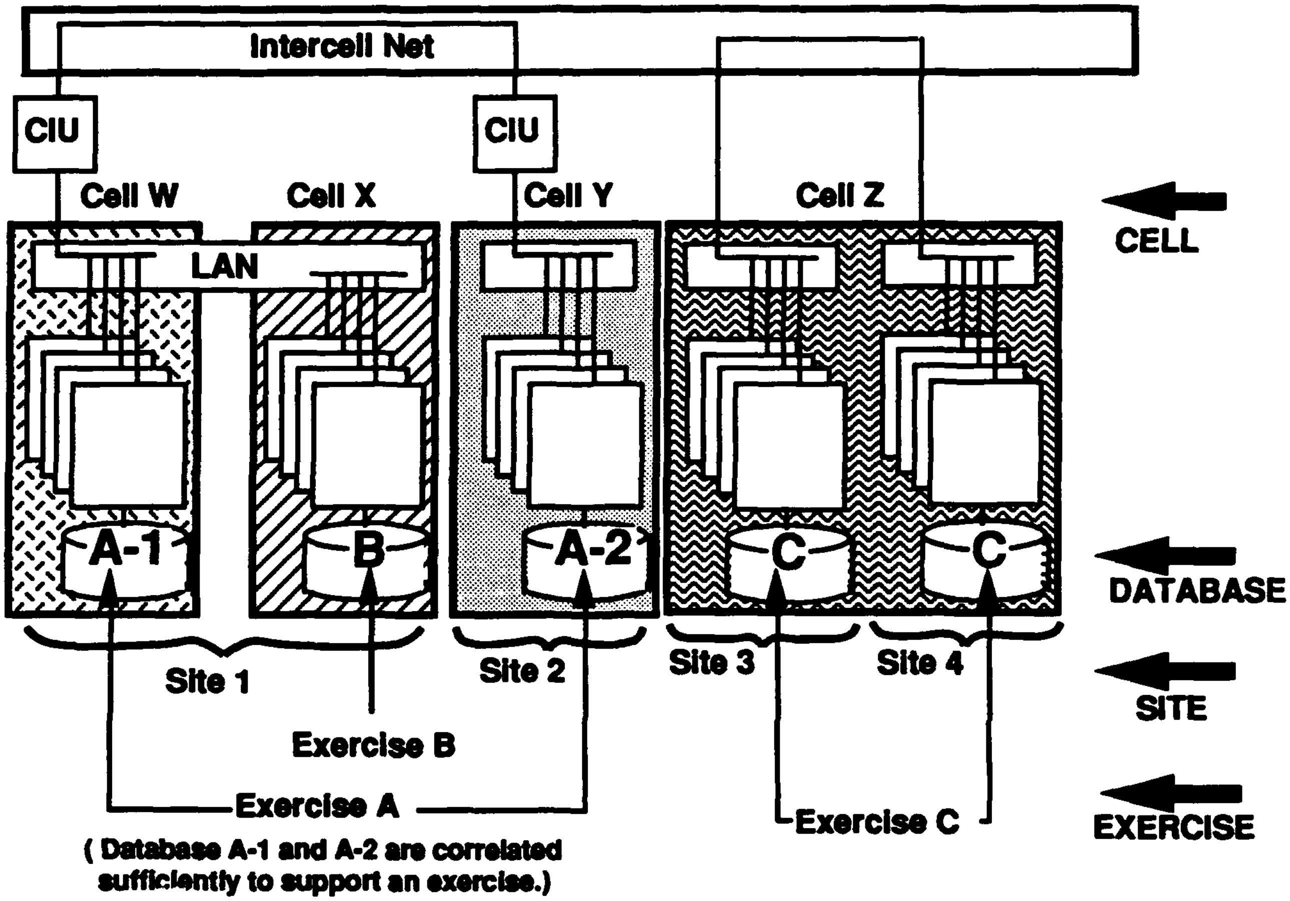


Figure 3.8-1 A multi-cell, multi-site, multi-exercise example.

4 Standards And Libraries

This section addresses the DIS message standard, the proposed DIS data base standard, and other related standards and libraries. It also touches on the issue of interoperability as it relates to standards.

4.1 DIS Message Standard

SIMNET established a protocol for communications between simulation entities over networks; see "The SIMNET Network and Protocols", Report No. 7627, June 1991. That initial protocol has been refined, enhanced, and published as a draft Military Standard for DIS -- "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation", IST-PD-91-1, October 31, 1991. The draft Military Standard has been submitted to the IEEE for acceptance as an application layer protocol standard and will also be submitted to the ISO standards committee for acceptance as an international standard.

The draft standard was developed with the full participation of the DoD simulation community and is already well understood by the community at large. It was recently made a requirement for the Close Combat Tactical Trainer (the first large scale production networked training device based on the SIMNET experience) and is being implemented as part of several ADST programs: the Rotary Wing Aircraft (RWA) program, the Crew Station Research and Development Facility (CSRDF)/AirNet Integration program, and the MultiRad program.

The draft standard includes both a required basic set of Protocol Data Units (PDUs) supporting combat interactions and a suggested interim set of PDUs addressing simulation control and additional battlefield environment simulations. However, the initial draft standard needs to be enhanced and expanded to incorporate the changes necessary to fully support evolving increases in simulation fidelity and capabilities for the DIS Architecture.

Simply adding or expanding PDUs to include ever increasing amounts of information about the simulated entities, the battlefield environment, and the simulation environment will require network bandwidths that will be unrealizable in practice for the foreseeable future. Therefore, the extensions to the PDUs must be coupled with definitions in the DIS Common Database (DIS CDB) to allow simulation cells and entities to compute information using local copies of standard algorithms and databases rather than passing all information across the networks. This fundamental paradigm is recommended as the basis for continued DIS PDU development.

The following sections briefly review the current definition of PDUs in the draft standard and several potential expansions to that standard to support DIS concepts without imposing unreasonable network loads. For the most part, our recommendations are entirely consistent with the body of work contributed by the

DIS community at large. However, we do offer some specific recommendations regarding the future direction of PDU development:

a. PDUs should be designed in concert with a data base, which should also be subject to the same standardization process.

b. New classes of PDUs will be required to handle the future inclusion of instrumented ranges, aggregated forces and Computer Generated Forces.

c. "Wrapper" PDUs will be needed to accommodate non-DIS messages that need to use the DIS network, including (possibly) exercise initialization message traffic.

d. As a general rule, PDUs should not be added if the desired result can be obtained by the use of common data bases (i.e., common SIMWORLDs). If PDUs are added, they should be designed to utilize a minimum of network bandwidth by sending only that data which is changing.

4.1.1 Current Status

The draft DIS PDU Standard defines a set of Protocol Data Units (PDUs) that form the current standard for communications between entities in a networked simulation environment. This draft standard is based on the SIMNET applications layer protocol with modifications made based on industry-wide participation in the DIS Standards Conferences held over the last 3 years under the auspices of PM TRADE and the Institute for Simulation and Training (IST).

This draft standard in its current form focuses on the basic interactions between simulated entities on the virtual battlefield. It does not establish requirements for PDUs related to other aspects of the simulation such as network management or terrain changes. It also focuses on the visual aspects of the land battlefield leaving the water, high altitude, and non-visual electromagnetic spectrum aspects of the battlefield for future enhancements to the protocol. Some aspects of these latter areas are addressed in interim PDUs in the draft standard.

Table 4.1.1-1 lists the PDUs identified by the current standard, their triggering events, and their destination. Table 4.1.1-2 provides the same information for the interim PDUs identified in the standard.

PDU	Triggering Event	Destination
Entity State	a) actual vs REA position exceeds threshold b) change in entity's appearance c) timeout or 5 sec has elapsed	(all)
Fire	a) moment that a weapon is fired	(all)
Detonation	a) moment that a munition impacts or detonates b) sky shot	(all)
Service Request	a) in need of logistic support	Servicing Entity
Resupply Offer	a) supplying entity receives a Service Request PDU for resupply request	Requesting Entity
Resupply Received	a) requesting entity receives supplies	Servicing Entity
Resupply Cancel	a) supplying entity cancels transaction b) requesting entity cancels transaction	Requesting Entity or Servicing Entity
Repair Complete	a) upon completion by repairing host	Requesting Entity
Repair Response	a) requesting entity receives Repair Complete PDU	Servicing Entity
Collision	a) collides with an object or another entity b) another entity collides into ownship	Colliding Entity

(Note: REA = Remote Entity Approximation, formerly the Dead Reckoning Algorithm.)

Table 4.1.1-1 Standard DIS PDUs

Interim PDU	Triggering Event	Destination
Activate Request	a) host computer intends to activate an entity at start of an exercise b) host computer intends to activate an entity in an exercise already in progress c) host computer is reactivating an entity that has been destroyed d) host computer is re-initializing an entity	(all)
Activate Response	a) entity receives Activate Request PDU	MCC
Deactivate Request	a) entity intends to withdraw from the simulation exercise b) MCC informing other entities of its intent to request an entity deactivation	(all)
Deactivate Response	a) entity receives a deactivate request	MCC
Emitter	a) entity's actual vs REA position exceeds threshold b) upon change in emitter mode c) timeout has elapsed	(all)
Radar	a) upon change in system mode b) upon change in power c) upon change in angles describing the volume of the scan d) upon change in the entities illumined	(all)

(Note: MCC = Management, Command and Control)

Table 4.1.1-2 Interim DIS PDUs

4.1.2 New PDU Characteristics

New PDUs and expanded PDUs, while responding to the need for conveying more information both about the entities and their characteristics, must be defined in concert with the proposed Common Data Base (CDB) Standard to minimize the amount of network bandwidth needed to support use of the PDU. The intent is to pass only changing information that cannot be computed by the receiving entity using standard algorithms and databases.

The concept of limiting the network load through judicious use of "change only" PDUs and local computation is a natural extension of the remote entity approximation approach for entity states and positions pioneered in SIMNET. The draft DIS protocol has already extended the REA concept by carrying REA identifiers in the Entity State PDUs, which are subsequently used by receiving entities to select the indicated REA from its local data base. Continued extensions of this PDU concept will play a significant role in supporting the anticipated growth in the number of entities participating in DIS, in supporting high fidelity electro-optical and acoustic simulation, and in using long haul networks to combine simulation resources.

The need for this type of approach can be illustrated by looking at the Electronic Warfare environment that is playing an increasingly critical role in the modern battlefield. The approach described below for EW can be used as a general model for other types and classes of PDUs. To a large extent, the ideas expressed below are implied by the current draft form of the interim EW PDU.

Precise simulation of electronic warfare signals at the emitter would require generation of a large amount of data on the signal structure and the spatial/temporal characteristics of the beam. This would necessitate use of a significant amount of network bandwidth to convey the information to the signal receiver. However, the emitter-receiver interactions in electronic warfare are typically long, complex interactions which can be simulated by defining action, initiation time, and electronic dead reckoning. Thus, a high degree of realism in electronic warfare simulation can be achieved by defining a PDU that conveys a basic set of emitter characteristics to potential receivers. The receivers, in turn, would then simulate the effect of the emissions on their sensors using local algorithms and databases that are characterized and defined in the CDB.

Unique PDUs would be used to convey information from platforms that are separate from PDUs used to convey the platform's visual appearance. For example, RF emissions would be assigned to a different PDU than the Entity State PDU. This will allow a vehicle to update its position without also generating the extra traffic associated with emitter parameters and emission characteristics that have not changed.

Additionally, the computations associated with certain simulations would be distributed across the network elements. For example, the effects of a radar illuminating a target may be simulated by the target, rather than by the emitter.

The emitter would indicate only the basic radar parameters and mode along with the fact that the radar was turned on. The target would then determine the local intensity and characteristics of the radar signal to determine its own sensory response. This approach will minimize the amount of network traffic while permitting high fidelity simulation of EW effects.

In summary, the following general guidelines for PDU development are suggested:

a. PDUs should be designed in concert with a data base, which should also be subject to the same standardization process.

b. As a general rule, PDUs should not be created if the desired result can be obtained by the use of data bases and agreed to playing rules (i.e., common SIMWORLDs).

c. New and modified PDUs should be designed to utilize a minimum of network bandwidth by sending only that data which is changing.

d. Fixed data should, in general, be transmitted prior to the start of the exercise via a standardized initialization process. This could take the form of media such as magnetic tape or other standard networks.

e. A natural consequence of (c) is the possible separation of component data (e.g., sensor data) from the host entity (e.g., the vehicle), thus creating two or more PDUs for certain classes of entities. Each PDU is then updated at the rate appropriate to that PDU.

4.1.3 Potential Extensions

The PDUs defined in the current draft standard are not sufficient to provide the full range of network communications necessary between entities in the BDS-D architecture. Specifically, they do not cover the initialization and control of exercises of heterogeneous simulation entities operating in multiple simulation cells connected by local, long haul, and wide area nets. Furthermore, they do not cover the increased fidelity anticipated in the battlefield environment including electronic warfare, weather, weapons effects, and dynamic terrain. Most of these requirements have already been identified by the DIS community and work is in progress. The following list tabulates the major categories of interest:

a. Communication PDUs (digital voice)

Although digital voice will introduce significant additional traffic onto DIS networks, communication PDUs are a necessary first step to the eventual inclusion of automated speech in DIS.

b. Emitter PDUs (Electronic Warfare, Radar)

The Emitter PDU is one of the more challenging design tasks, as discussed above. Data base oriented approaches are recognized as a necessity by the PDU designers. Some thought is being given to extending the emitter PDU to the undersea domain.

c. Exercise Control PDUs

The intent here should be to keep the number of new PDUs to a minimum, making maximum use of data base mechanisms to initialize the exercise. Initialization will need to consider network topology, simulation and simulation support entities, and perhaps rendering/fidelity controls for the visual systems involved in the exercise.

d. Weather/Atmospheric PDUs

New PDUs are needed to address the dynamic effects of weather, such as moving storm cells, and man-made atmospheric effects such as chaff and smoke.

e. Dynamic Terrain PDUs

A dynamic terrain PDU design was put forward at the last DIS conference. It suggests the need for several PDUs to accommodate the variety of effects introduced by dynamically changing cartographic data, such as bomb craters, earthworks (berms, trenches), and damaged/destroyed buildings.

f. Munition PDU Extensions

The current standard could be expanded to include submunition carriers such as WAM (air delivered mines) and mixed submunition carriers such as DAACM (which was designed to carry both mines and explosive penetrators). A shoot-to-kill category for explosively formed penetrators (EFPs) might be added under the high explosive category since they function somewhat differently than conventional hit-to-kill shaped charge devices. Another category that might be added is concussion bombs such as FAE (fuel air explosive) bombs which are not truly high explosive devices.

With proper warhead identification, the actual PDU messages transferred during a wargame can be streamlined considerably without loss of quality. Many of the characteristics contained in the fire and impact PDUs could actually be handled locally by the simulation entities, rather than being continually broadcast by the firing entity.

g. Instrumented Ranges PDUs

A new class of PDUs is needed to accommodate instrumented ranges, where network bandwidth is at a premium.

i. Aggregated Forces PDUs

A new class of PDUs is needed to support the migration of DIS from platform-only interactions to aggregated force interactions.

j. Wrapper PDUs

Wrapper PDUs would provide the ability to encapsulate non-DIS messages that need to use DIS network services. For example, exercise initialization commands, or data base downloads that precede the actual exercise might use wrappers.

k. Computer Generated Forces PDUs

Finally, it is anticipated that as a CGF architecture evolves for DIS that a new class of PDUs will be required.

4.2 DIS Data Base Standard

4.2.1 Overview

The DIS Architecture is intended to connect many heterogeneous simulators together on a common, simulated battlefield to engage in a single, combined exercise. The proposed DIS Data Base Standard, along with the DIS Message Standard, will constrain the architecture reference model to support the interoperability goals for DIS. This is shown conceptually in figure 4.2.1-1.

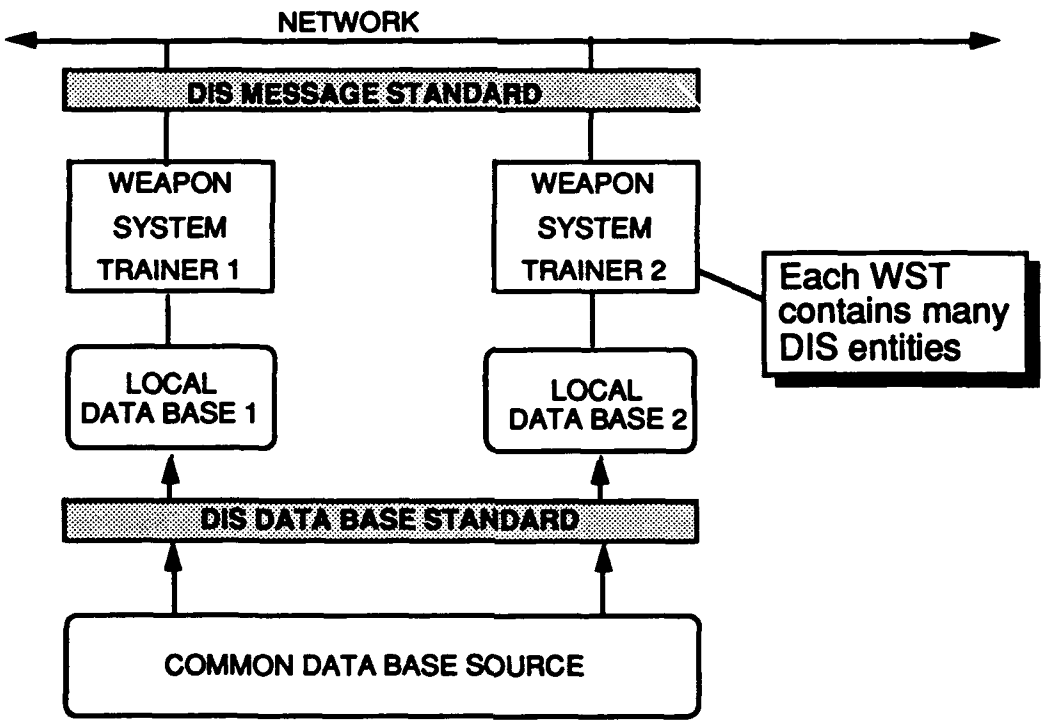
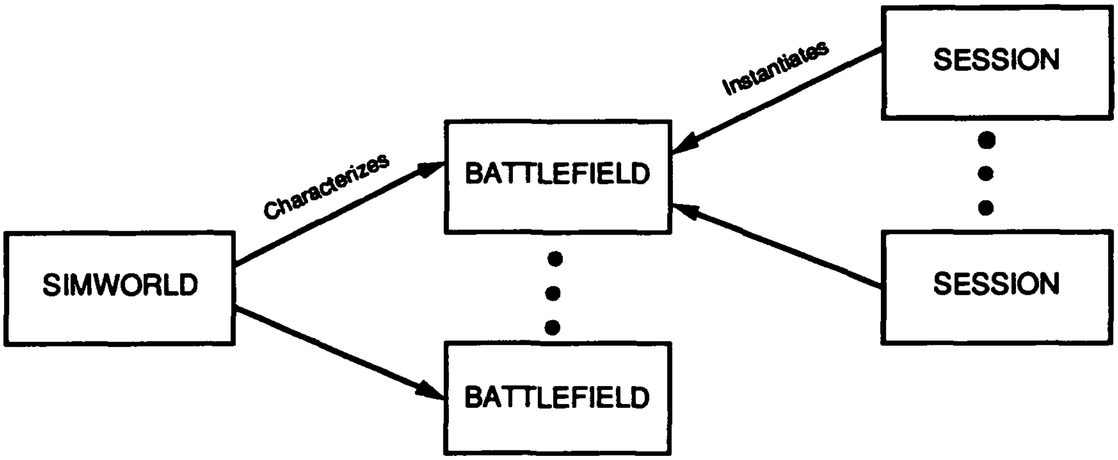


Figure 4.2.1-1: DIS Data Base and Message Standards

The DIS CDB Standard described herein divides the DIS world into three major categories: SIMWORLDs, BATTLEFIELDs, and SESSIONs. SIMWORLDs define the underlying models of the BATTLEFIELD: the remote entity approximation algorithms, the atmospheric models, the terrain models, weapons and weapon effects models, rendering algorithms, and so forth. BATTLEFIELDs consist of the gaming area and the geometry and attributes of the fixed and moving components that comprise it: the terrain, cultural features and models that reside on the terrain, air, land and sea vehicles, weapons, sensors, and the environment. SESSION Data Bases define initial and dynamic exercise conditions including weather and dynamic terrain effects, network topology, and player(entity) identification and positioning. SESSIONs provide the glue that binds together all of the elements that comprise a DIS Exercise. Figure 4.2.1-2 shows the relationships between the database components that are to be defined by the DIS Common Data Base Standard.



A *Simworld* characterizes a Battlefield. Several different *Battlefields* can have the same *Simworld*. Several *Sessions* can provide unique instantiations of a *Battlefield*, i.e., they can set up different battle scenarios on the same geography with different players and environmental conditions.

Figure 4.2.1-2: Relationship Between DIS Data Base Components

The proposed DIS CDB is defined as a hierarchical collection of data bases, consistent with the evolving DIS Architecture. Figure 4.2.1-3 illustrates the hierarchy.

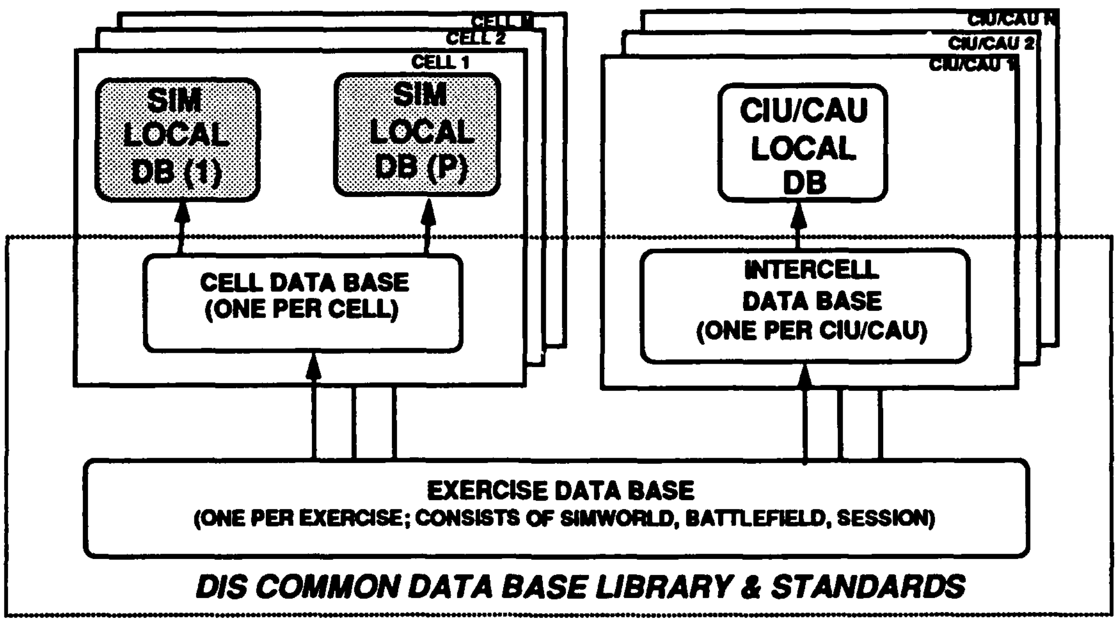


Figure 4.2.1-3 : DIS CDB Hierarchy

Referring to Figure 4.2.1.3, the Exercise Data Base is shown at the bottom of the hierarchy. The Exercise Data Base is defined as all of the data base components needed to perform the given exercise. As shown, an exercise can span multiple cells, where each cell is defined as a homogeneous collection of simulation entities. Each cell has associated with it exactly one "cell" data base - the SIMWORLD, BATTLEFIELD and SESSION data bases used by all simulation entities in that cell. Each simulation entity within the cell has its own local or private data base which is derived from the cell data base. The local data bases in each cell will tend to be different from simulator to simulator, but all derive from the same cell data base.

To illustrate this concept further, Figure 4.2.1-4 is provided. It shows the end-to-end data flow of the DIS data base generation process. A DIS Exercise Data Base is created by drawing on DIS Libraries of SIMWORLD Specifications, BATTLEFIELD Data, and SESSION Data. The Exercise Data Base is the recommended configuration control point for the overall process. The next step is the partitioning of the Exercise Data Base into Cells, as shown. Note that an Intercell Data Base is created; it can be viewed as the union of all cell data bases that are used for the given exercise. From this union, the individual CIU and/or CAU local data bases are developed. In a similar fashion local data bases are created for each simulation entity in each cell. This is shown as a transformation process, since in the case of terrain data it is converted from a grid form to a polygon form, and all data types are merged and formatted into hardware loadable form. All local data bases are private data bases, since they tend to be driven by the architecture of the host simulation entity. On the far right of the diagram a real-time view of the DIS Architecture is shown.

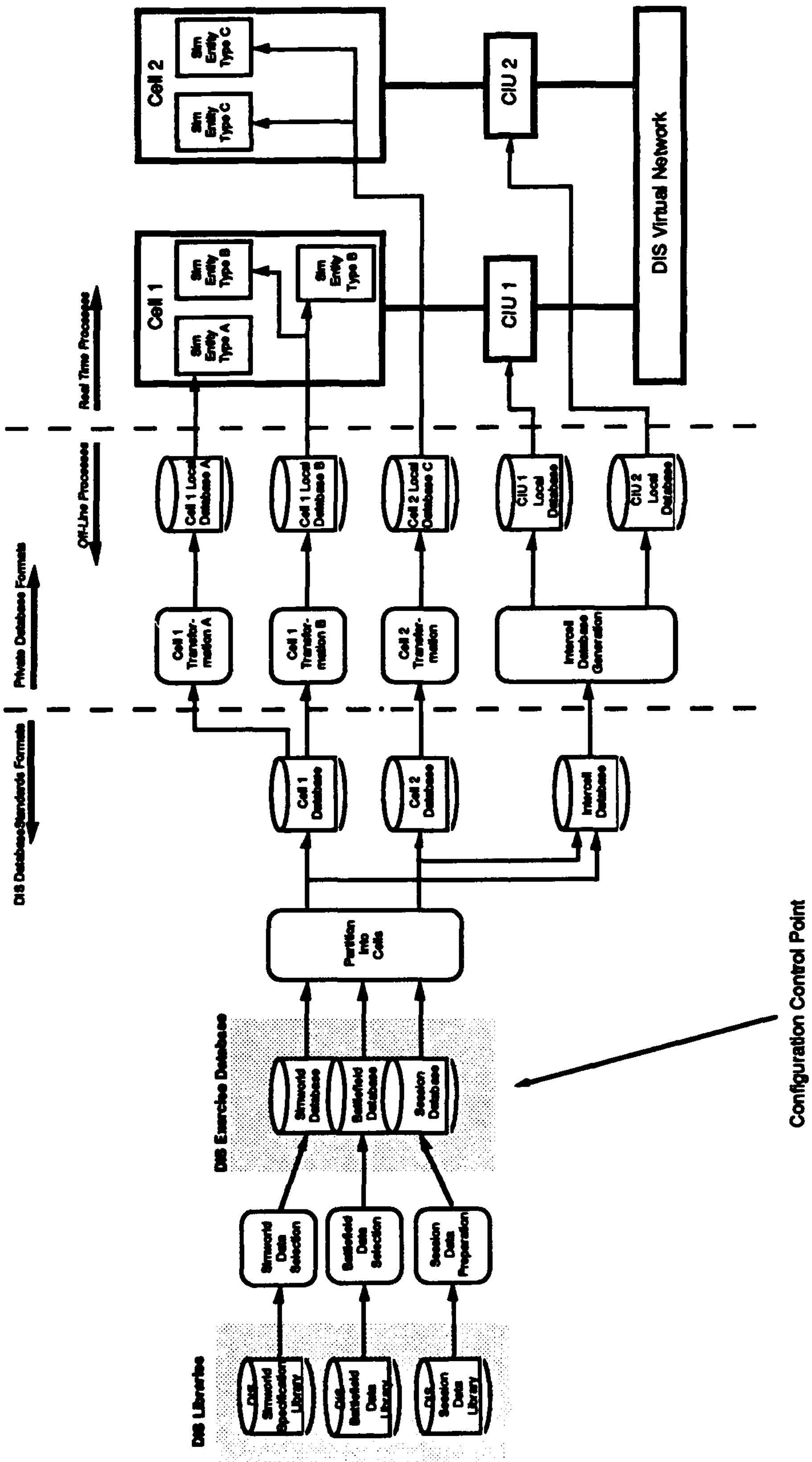


Figure 4.2.1-4: DIS Data Base Generation Concept

4.2.2 Data Base Contents

The DIS CDB Standard described herein divides the DIS world into three major categories: SIMWORLDs, BATTLEFIELDs, and SESSIONs. SIMWORLDs define the underlying models of the battlefield: the remote entity approximation algorithms, the atmospheric models, the terrain models, weapons and weapon effects models, rendering algorithms, and so forth. BATTLEFIELDs consist of the gaming area and the geometry and attributes of the fixed and moving components that comprise it: the terrain, cultural features and models that reside on the terrain, air, land and sea vehicles, weapons, sensors, and the environment. SESSION Data Bases define initial and dynamic exercise conditions including weather and dynamic terrain effects, network topology, and player (entity) identification and positioning. SESSIONs provide the glue that binds together all of the elements that comprise a DIS Exercise. Figure 4.2.2-1 illustrates the proposed organization of the DIS CDB. Since in general there is a one-to-one correspondence between items in the SIMWORLD and BATTLEFIELD data bases, the distinction between them is not shown, rather the data elements are described as horizontal slices (SIMWORLD and BATTLEFIELD data combined).

DATA BASE ELEMENT	TYPE/CHARACTERISTIC	PDUS
Cartographic Data	**Static Data**	NO
Terrain	Gridded terrain data	
Culture	Points, lineals, areals	
Models	Geometry, attributes	
Texture	Imagery	
Platform Data	**Entities**	YES
Vehicles	Geometry, appearance, dynamics, articulation, kinematics	
Lifeforms		
Sites (relocatable)		
Munition Data	**Entities**	YES
Guided	Geometry, appearance, dynamics, kinematics	
Non-Guided		
Environment Data	**Entities**	YES
Weather	Fog, lighting, TPH, wind	
Atmospheric Effects	Smoke, dust, chaff, flares	
Dynamic Terrain	Craters, berms, buildings	
Electromagnetic Data	**Components**	YES
Visuals	Rendering, load management	
Electro-Optical	FLIR, NVG, LLLTV	
Radar	Ground mapping, SAR, TFR	
Electronic Warfare	Elint, jammers, C3I	
Radio Nets	Digital Voice Communications	
Session Data	**Control Data**	YES
Network Initialization	Topology	
Entity Initialization	Position, attitude, stores, etc.	

Figure 4.2.2-1: Proposed DIS CDB Organization

4.2.2.1 SIMWORLD And BATTLEFIELD Data Bases

The SIMWORLD and BATTLEFIELD Data Bases are organized in similar fashion. For each model or algorithm in the SIMWORLD Data Base there is, in general, a corresponding data type or entity in the BATTLEFIELD Data Base. Five major categories of models and entities are defined: Cartography, Platforms, Munitions, Electromagnetic, and Environmental. Each is briefly described in the following paragraphs.

4.2.2.1.1 Cartography

The cartographic models in the SIMWORLD Data Base describe the underlying structure of the terrain, culture, 3D models, and texture data that resides in the BATTLEFIELD Data Base; these data correspond to real world features in a gaming area described by the actual geography (e.g., North Korea).

Terrain The terrain model is defined as a regular grid (or grids) of elevation posts of specified accuracy and resolution (e.g., 3 arc sec spacing, +/- 30 meter accuracy, 1 meter resolution). The terrain data itself consists of elevation grid posts of the prescribed grid spacing and accuracy for the specified gaming area.

2D Culture The culture model is defined as a collection of point, lineal, and areal feature models that describe the form, accuracy, fidelity and attributes (color, texture, feature type) of the culture entities. Included also are terrain attributes, such as slope and electromagnetic properties. The entities, in turn, are real world geographic features expressed in vector and polygon form as point features (references to trees, towers, buildings, but not the actual features), lineal features (roads, rivers), and areal features (forested areas, large buildings).

3D Models The 3D Models are defined in the SIMWORLD Data Base as Polygonal or Constructive Solid Geometry (CSG) models with specified accuracy, fidelity and attributes (color, texture, feature type). The BATTLEFIELD Data Base contains the physical description of the real world 3D Models that are included in the specified gaming area - the trees, buildings, towers, and all other man-made and natural 3D features that are fixed on the terrain's surface.

Texture The texture defined in the SIMWORLD Data Base is an image raster with specified accuracy and resolution (e.g., 10 meter spacing, +/- 10 meter accuracy, 8 bits per pixel resolution). The texture described in this category corresponds to ground (terrain) texture, as opposed to the 3D Model texture included with the 3D Model category.

4.2.2.1.2 Platforms

The platform models in the SIMWORLD data base describe the basic motion models for the moving entities which reside in the BATTLEFIELD data base. In addition to the vehicle simulations with their motion models, other types of platforms include lifeforms, and sites (sites are stationary platforms).

Vehicles The vehicle models include not only their motion and performance models described under dynamics and kinematics below, but models of the equipment fit as well. The vehicles are the entities which provide a coordinate location point for all equipments which may be included (e.g., radio, sensors, detection devices, weapons, countermeasures devices, etc.). The models must also include signatures, including passive visual, IR, radar cross section as well as unintentional emissions, and vulnerability. A part of the model is, of course, the vehicle size and shape to determine when interaction with terrain or other vehicles occurs.

The dynamics and kinematics which control the action of the vehicles, include propulsion effects, effects of environment (air/terrain/water), control equations, guidance laws for automatic control features, performance capabilities and limit controls, and vehicle characteristics such as drag and buoyancy. Depending on vehicle type, these may vary from 3 DOF to 6 DOF (degree of freedom) parameters with acceleration performance in all axes. Where appropriate, articulation effects would be included. Unmanned or semi-automated vehicles would include additional data such as tactics, scripted actions, and command/control links; some of this data may be stored in the SESSION data base.

Lifeforms Lifeform models are similar to vehicle models. They represent the basic action of the lifeforms, including: propulsion effects, effects of environment (air/terrain/water), control equations, guidance laws for automatic control features, performance capabilities and limit controls, and physical characteristics such as drag and buoyancy. Where appropriate, articulation effects may be included. Unmanned or semi-automated lifeforms would include additional data such as tactics, scripted actions, and command/control links; some of this data may be stored in the SESSION data base.

Sites Site models are similar to vehicle models in that sites may represent collections of equipment in a physical space. As such, site models must include signatures (passive visual, IR, radar cross section as well as unintentional emissions), and vulnerability. Unmanned or semi-automated sites would include additional data such as tactics, scripted actions, and command/control links; some of this data may be stored in the SESSION data base. An example of an unmanned site would be a SAM site that reacts to the presence of aircraft by turning on detection and tracking radars and launching surface to air missiles.

A site is the stationary equivalent of a platform. Examples are Command and Control Sites, *Air Defense Sites (like the SAM site referenced above)*, and EW Sites consisting of radars, Elint devices, and jammers. The value of a site lies in its ability to provide position and appearance data and to be moved or defined as a unit in an exercise.

4.2.2.1.3 Munitions

The munitions models must contain most of the elements covered for vehicle above, as appropriate, including command and control links. In addition, models for guided munitions (whether command guided, active seeker, passive seeker, or semi-active) must include the guidance model and models of the seeker system including considerations of target motion and maneuver, electronic warfare and weather. Munition impact calculations are included.

The dynamics and kinematics for munitions are similar to those for vehicles, except that ballistic munition models would be included as appropriate, controls are generally simpler, and articulation is normally not an issue.

4.2.2.1.4 Electromagnetic

Desert Storm has clearly shown that electromagnetic considerations are critical to much of future warfare and is, therefore, critical to both advanced simulation and to DIS. Work done to date on other simulation programs such as SOF ATS has demonstrated EW to be one of the most critical drivers in the development of data bases and PDUs. EW requires the highest data rate if a brute force approach is taken; EW also requires the highest degree of interaction since each platform generally contains several electromagnetic elements which may potentially interact with elements in every other platform. In addition, each EW platform has an "appearance" characterized at a number of frequencies in the EM spectrum; the terrain has EM characteristics of importance; the atmosphere has EM properties which must be characterized at every point in the three dimensional space that surrounds the battlefield.

EW Emitters The Data Base in this area includes the electronic combat environment consisting of RF and IR elements, communications elements and the C3 net. Together, these models would be organized into entities comprising a site or vehicle with one or more elements and linked via a command and control structure. Each entity and significant subelement of each element would be controlled by tactics which are part of the SESSION data base and are related to communications. The C3 net would also include tactics as well as alternate paths/tactics as a result of changes in the net (destruction of nodes, instructor commands, etc.). Likewise entity tactics must include tactics appropriate to the level of connectivity to the C3 net and to specific commands from the C3 net. Some specific command tactics would reside in the SESSION data base.

The RF models contain sensors (radiation, detection of returned signal), intercept equipment (i.e. elint, warning receivers, etc.), jammers, and passive models (i.e. chaff, signatures) which, in high fidelity simulators, are interactive with other simulation entities and which have high levels of interaction with terrain and weather. Chaff also has the characteristic of being separated from the entity and, in high fidelity models, having a trajectory of its own. These models are characterized by their complexity, their impact on a large number of entities, and their extreme range of time effects (even excluding pulses they include rapid scans and jamming effects as well as very long EW processes).

The C3 models are probably the most classified. They are extremely critical to a realistic EW simulation since they establish connectivity, alert and control of entities, and also affect the communications and EW simulations of high fidelity simulations. C3 models operate between entities and within some of the more complex entities (i.e. those with more than one sensor or weapon system).

In the individual simulator, algorithms for the EW equipment are required. These algorithms determine the impacts of the variability of sensed signals received due to sensor scanning and other effects, which signals will be detected, what response (visual, audio, jamming) would occur is detection results, and the details of all resulting EW activity. These are peculiar to the specific equipment being simulated and to the fidelity requirements imposed at the time of simulation. The required data base must adequately describe the fidelity and response characteristics to support interoperability.

Radar Models for these sensors are specified in the SIMWORLD data base. Elements of the models are then used in the entities in the combat simulation. Models include significant radiation characteristics (power, signal characteristics, antenna/illuminator pattern details, etc.), detection, EW (ECM, ECCM, vulnerability to ECM), timing, etc. As will be discussed, these models must be specified so as to limit the bandwidth required for PDU traffic which requires that calculations be spread between the simulator with the sensor and the simulator with the target. Hence, key elements of these models must be shared in a distributed fashion.

Radio Nets Models for the communications nets and communication devices will be defined in the SIMWORLD Data Base. These definitions will include the level of simulation fidelity, fidelity of the terrain and intervening object signal occulting models, accuracy of the jamming and interference models. The models used in the radio simulation will be included in the BATTLEFIELD database and the specific parameters for the exercise will be in the SESSION Data Base.

OTW Visuals and EO One of the most difficult tasks is the development of CDB standards and PDUs for OTW visual systems. Even identical visuals can behave differently if the load management algorithms in the different systems are not uniformly applied. For example, one IG could be set to maximize close-in detail at the expense of distant cues, and vice versa for the other IG.

One approach to this problem is the use of identical polygon budgets for each feature type for each visual system. These budgets would then be modified according to the efficiency with which each visual system was able to render each feature type. Efficiency measures would be defined in the SIMWORLD; real-time changes in polygon budgets (properly weighted) would be defined via PDUs or via pre-defined rules stored in the local data bases.

IR IR models are similar to the RF models except that the level of technology limits their complexity and time variability. There are sensors, warning receivers, jammers, and flare models. Flares, like chaff, have a trajectory. In addition, the emissions of entities in various frequency bands (IR, Visual, RF, etc.) must be described in such a way that they can be represented by the different sensors being simulated. For example, entities in a night scene within view of an image generator simulating the view through night vision goggles (NVGs) or low light level television (LLLTV) have to be represented with parameters that would result in a correct depiction in the simulated scene.

4.2.2.1.5 Environment

This section of the data base specifies and defines weather, atmospheric effects, and dynamic terrain.

Weather and Atmospheric Effects High fidelity simulators will require a time scripted data base covering the entire gaming area in some detail. This data base must include the effect of the weather at each point in space on signals at several frequencies in the electromagnetic spectrum. This data is used in high fidelity simulators to compute the detailed path loss for RF, IR, and EO signals. The weather data base must also support missile flyout models and vehicle motion models with aerodynamic parameters. In addition the weather model will, for high fidelity simulators, include the sun and moon and their location with time (for light and, in the case of the sun, for its effect of IR seekers). Other weather-like effects, such as battlefield smoke screens and dust clouds and biological weapon effects, are handled in a similar fashion.

Dynamic Terrain Dynamic terrain effects will likely be defined parametrically in the SIMWORLD. For example, berms and craters could be represented as trapezoidal volumes with positive and negative elevations, respectively. Collateral damage might be represented by "kill" or "damage" codes attached to the affected data base feature. Dynamic terrain is discussed further in Volume II of this document.

4.2.2.2 SESSION Data Base

The SESSION Data Base is conceptually different from the SIMWORLD and BATTLEFIELD databases in that it contains detailed information related to a specific exercise or SESSION. The SESSION Data Base portion of the standard defines its contents in two major categories: Network data and Simulation Entity data. Each is briefly described in the following paragraphs.

4.2.2.2.1 Network Initialization and Control Data

This section of the database includes the information needed to set up and operate the simulation network to connect all of the cells and entities involved in the exercise.

Routers, Gateways, Bridges This data provides the logical and physical mapping for the virtual network, both the Wide Area and Local Area tiers, as defined in the BDS-D architecture. It provides the information needed to set up and establish the connections between the Cell Interface Units and between the entities in the local simulation systems.

Cell Interface/Adapter Unit This data provides the information to allow the Cell Interface/Adapter Unit to set up for the exercise including defining the filtering that will be done to manage the level of network traffic. It also provides the protocol conversion information, if necessary, to handle the interface to non-standard cells.

Cell Initialization and Control This data provides the information needed to configure the cell to support the exercise.

4.2.2.2.2 Simulation Entity Initialization and Control Data

This section of the database includes the information needed to initialize all of the simulation entities with the parameters and data for the specific exercise. It includes initial conditions and control measures.

Manned Battlefield Entities This data provides the information needed to configure each individual simulator and its local environment to support the exercise. It includes the software, hardware, and database configuration of the simulator, the models to be used, the level of fidelity required, initial parameter values, and the initial state of the battlefield.

Unmanned Battlefield Entities (Computer Generated Forces) This data provides the information needed to configure the unmanned battlefield entities supporting the exercise. It includes the software, hardware, and database configuration of the Computer Generated Forces engine, the forces to be controlled at each CGF node, the models to be used, the level of fidelity required, initial parameter values, and the initial state of the battlefield.

Environment Entities This data provides the information needed to set up and initialize environment entities. It includes the models to be used, the level of fidelity required, initial parameter values, and the initial state of the battlefield.

Support Entities This data initializes and controls the support entity devices such as After Action Review (AAR) stations, Master Control Consoles MCC), and Maintenance Consoles. AAR station data includes the identification of the stations to be used for observing and recording the exercise data, the software, hardware, and database configuration of the AAR stations, and any preset observation and recording parameters such as observation positions, key event marks, and special logging requests. MCC data includes the identification of the exercise file, dynamic changes to be made during the exercise, and control parameters to be monitored. Maintenance Console data includes the identification of the equipment being used in the exercise, special diagnostics that must be run, and parameters to be monitored.

4.3 Related Standards And Libraries

Clearly there is a need for a standard way of dealing with the new DIS data base as it evolves. As stated in the BDS-D Program Plan:

> An open system design architecture, with a common set of protocols and standards, to achieve interoperability of simulations, will be the keystone of the program development.

At the same time, there is no need to reinvent new standards if existing standards can be used in whole or in part. The following paragraphs discuss existing data standards and their relevance to DIS.

4.3.1 Cartographic Standards

The terrain or cartographic portion of the new DIS CDB standard could be modeled after one of the existing standards from DMA or Project 2851, or even a commercial format, such as Software System's MultiGen Flight Format. These alternatives are tabulated below in Table 4.3.1-1 and discussed in the following paragraphs.

EXISTING STANDARD	TERRAIN	CULTURE (2D)	MODELS (3D)	TEXTURE (IMAGES)
DTED, DFAD (DMA)	{3, 1} ARC SEC GRIDS (DTED)	VECTOR (DFAD)	---	---
ITD (DMA)	3 ARC SEC GRID	VECTOR (*)	---	---
SSDB, SIF (P2851)	VARIABLE GRID	VECTOR	POLYGONAL & CSG	RASTER
GTDB (P2851)	POLYGONAL & GRID	POLYGONAL & VECTOR	POLYGONAL	RASTER
FLIGHT (MULTIGEN)	POLYGONAL	POLYGONAL	POLYGONAL	---

(*) Culture types: slope, vegetation, surface materials, surface drainage, transportation, obstacles.

Table 4.3.1.-1: Existing Cartographic Data Base Standards

4.3.1.1 DMA

The Defense Mapping Agency (DMA) has provided source data to the visual and radar simulation community for many years. The digital products that have been used most widely are Digital Terrain Elevation Data or DTED and Digital Feature Analysis Data or DFAD.

Digital Terrain Elevation Data (DTED) DTED provides elevation grids in 3 arc sec format (approximately 100 meter spacing at the equator) for much of the Western world and many hot spots. Absolute vertical accuracy is on the order of +/- 30 meters. For selected portions of the world 1 arc sec format data is supplied. The two formats are referred to as Level I and II, respectively.

Digital Feature Analysis Data (DFAD) DFAD provides cultural feature data in vector form in three classes: point features, lineal features, and areal features. Point features are references to three dimensional features such as power pylons, radio towers, buildings, water towers, and so forth. The actual 3D models are not supplied by DMA, rather they are drawn from a model library supplied by the visual or radar simulator supplier. Lineal features are vector descriptions of roads, railroads, rivers, runways, and other natural and man-made features that can be described by a centerline and a width. Areal features are polygonal features such as the footprints of large buildings, forested areas, farmlands, lakes, and other natural and man-made features that can be described as a chained vector outline. All of the point, lineal and areal features are attributed with characteristics such as feature type codes (FIC), surface material codes (SMC), key distances (width, height, length) and other data necessary to characterize the feature for simulation purposes.

DFAD is available in three versions, referred to as Levels I, II, and III. Within each version, different "editions" are available. Level I DFAD is the most commonly used version. It contains detail roughly equivalent to 1:250,000 scale JOG charts, and covers an extent of 1 degree by 1 degree. Levels II and III are designed to provide high resolution patches of culture data; patch sizes range from 2 nm by 2 nm to 8 nm by 10 nm. Detail on these maps is roughly equivalent to 1:50,000 scale maps or better. Absolute horizontal accuracy for the DFAD products varies; Level I DFAD is advertised at 80 to 90 meters.

Both DTED and DFAD are currently being supplied in WGS-72 or WGS-84 format. The difference in the two datums is very small, and it is a straightforward exercise to convert one into the other. Both the terrain and culture formats are well understood by virtually all visual and radar simulation suppliers.

Interim Terrain Data (ITD) Interim Terrain Data or ITD is a relatively new DMA product. It contains DMA DTED as one of its data types. It also includes surface feature data in vector form that is complementary to DFAD: surface configuration (slope), vegetation, surface materials, surface drainage, transportation, and obstacles. This data base standard is intended by DMA to support (quoting from DMA's "Digitizing the Future"):

> ...operations, intelligence, and logistics planners in the performance of automated terrain analysis tasks such as terrain visualization, route/site selection, mobility/countermobility planning, communication planning, navigation, and fire support planning and execution.

This data is particularly well suited to ground combat simulation applications.

4.3.1.2 Project 2851

Project 2851 is a tri-service program that has developed two simulator data base formats: the SSDB (Standard Simulator Data Base) Interchange Format (SIF) and the Generic Transformed Data Base (GTDB) format. Both formats are derived from a common internal source format called the Standard Simulator Data Base (SSDB). Project 2851 format compatibility has become a standard requirement in simulator RFPs. Visual systems are required to support both the input and output of SSDB data via the SIF format. It is expected that future DoD RFPs will require vendors to provide a data base in SIF format as a deliverable item, in addition to the IG specific local data base required to load into the host Visual System. This is consistent with Project 2851's new role as librarian for simulator data bases. As shown in figure 4.3.1-1, Project 2851 will accept new data bases from simulator vendors, merge the data with the existing library (which implies the resolution of inconsistencies), and store it as SSDB data. The user will then be able to request data in either SIF or GTDB form. Project 2851 will also support the creation of new data bases from raw source materials, but this mode of operation will be secondary to its role as librarian.

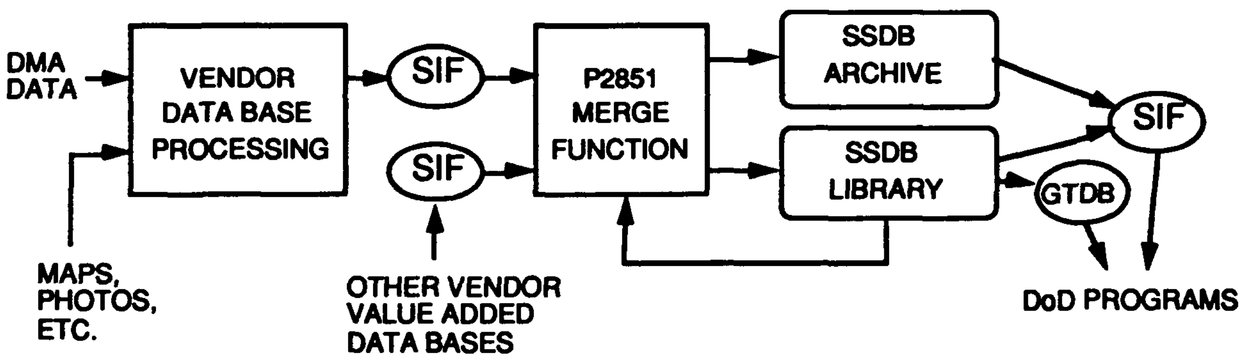


Figure 4.3.1-1: Project 2851 Library Concept

The GTDB format is strictly an output format, unlike the SIF which is bi-directional. In addition, the GTDB format supports transformation of the SSDB data into a polygonized, merged form, ready for formatting by the receiving simulator into a run-time or hardware loadable format. Some vendors prefer to use their own transformation techniques; therefore, GTDB data can also be requested in a non-transformed form, similar to SIF. This untransformed form of Project 2851 data, available in both SIF and GTDB formats, is selected by visual system vendors that prefer to utilize unique and often proprietary methods of transforming the "raw" input data into a format optimal for the vendor's image generator architecture.

Figure 4.3.1-2 illustrates a typical data base generation process for a visual simulation system. Note that the intermediate product in the process is very similar in structure to Project 2851's SSDB: terrain, culture, models and texture are stored in a geographically registered but non-merged and non-polygonized form. The output of the process is similar in structure to Project 2851's GTDB: the data elements are polygonized, merged and formatted into a private, local data base.

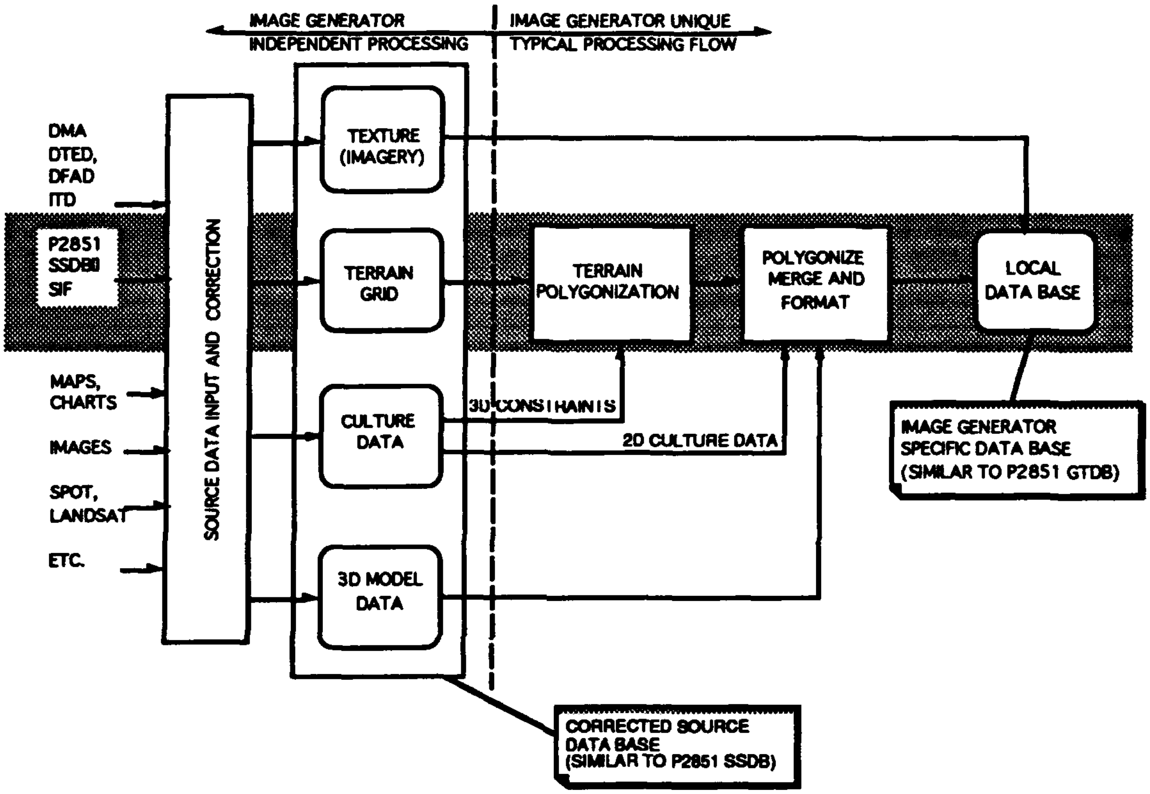


Figure 4.3.1-2 Data Base Generation for Visual Systems

The SIF is a comprehensive format that spans all of the major simulator data base categories - terrain, culture, models, and texture. No other candidate format has this breadth of coverage. In comparison with DMA's standard DTED and DFAD products, SIF is a superset, as shown in Table 4.3.1-2. The SIF overcomes the limitations of the DMA terrain and culture products, and adds model and texture formats. The new data types offered by DMA's new ITD product can be incorporated into the SIF format. In fact, Project 2851 was modified in August 1991 to accept ITD as another SSDB input data type.

FEATURE	DMA DTED, DFAD	PROJECT 2851/SIF
TERRAIN		
- GRID SPACING	{3, 1} ARC MIN	{3, 1, .1, .01} ARC MIN
- EXTENT	1 DEG x 1 DEG	USER SPECIFIED
- RESOLUTION	1 METER (16 BITS)	.01 METER (24 BITS)
CULTURE		
- TYPES	3 - POINT, LINEAL, AREAL	6 - ADDS POINT LITES
- ATTRIBUTES	LIMITED, FIXED	EXTENSIVE, EXTENSIBLE
- RESOLUTION	.1 ARC SEC (*)	.01 ARC SEC
- LOD'S	NO	{100, 30, 10, 3, 1} METERS
- 3D FEATURES	NO	YES (TERRAIN FEATURES)
- TRACEABILITY	NO	PROVISIONS MADE
GENERAL		
- MEDIA	9 TRACK TAPE, CD ROM	9 TRACK TAPE, 8 MM TAPE (UNDER STUDY)
- SUPPLIER(S)	DMA	PRC, INDUSTRY

(*) .1 Arc sec is approximately 10 feet at the equator

Table 4.3.1-2 Comparison of DMA and Project 2851 Data Formats

The SIF is currently in draft form as a new military standard. It was submitted to the government for review and approval in December, 1991. It appears to be an excellent model for the DIS Common Data Base Standard. We propose to simply extend the SIF format to meet DIS needs, making minimal changes to the existing SIF draft standard in a cooperative fashion with Project 2851.

The issue of whether to use GTDB or SIF data from Project 2851 is still being debated at the Project 2851 Industry/Service Working Group (ISWG) Meetings. Since the GTDB will be available in non-transformed form (like the SSDB), the issue is a moot point. That is, GTDB elevation data can be requested in grid from, and both the terrain and culture data sets provided as separate, non-merged data files. Based on the discussions at the last Project 2851 ISWG meeting held in Daytona Beach in late January, 1992, it appears likely that both the SIF and the GTDB will survive as Project 2851 output products.

Regardless of final output format, the user will benefit from the "value-added" nature of the Project 2851 system. As illustrated in Figure 4.3.1-1, the Project 2851 library concept will add value to simulation data bases at two levels:

- Each vendor will significantly enhance raw source products such as DMA DTED and DFAD, and provide the resulting value-added data base to Project 2851 in SIF format.

- Project 2851 will merge the vendor supplied data bases with existing library data bases, and retain the best features for subsequent distribution via SIF and GTDB data bases.

4.3.1.3 Other Cartographic Standards

A number of cartographic standards exist or are emerging in addition to the DMA and Project 2851 standards just discussed. However, none of them offer the breadth of coverage afforded by the Project 2851 standards. These standards and their shortcomings (relative to the recommended standard) are briefly described below.

DIGEST is the European equivalent of DMA DTED and DFAD; however, like the DMA products, it offers no support for models or LODs. The SDIS or SIMNET Database Interchange Specification is another cartographic standard, but it does not support terrain grid data or geo-specific texture, and uses a very rigid format that would be difficult to extend. The USGS is developing SDTS or Spatial Data Transfer Specification, but this is still in a development stage, and at this point there is no plan to support models.

DMA's emerging Vector Product Format or VPF was seriously considered by Project 2851, but it does not support models, texture, or gridded data. Even as a culture-only standard the VPF format is inefficient, due to its expression of culture data as non-overlapping objects. Thus, VPF is well suited to the map-makers and map-users, but the simulation community has tended to favor overlapping, layered objects, as currently implemented in DMA's DFAD and Project 2851's SIF formats.

Finally, it should be mentioned that the USAF is developing a new Common Mapping Standard or CMS, which in reality is a repackaging of a number of existing and emerging cartographic formats: ADRG (Arc Digitized Raster Graphics), ADRI (Arc Digitized Raster Imagery), DCW (Digital Chart of the World), DTED, DFAD, WVS (World Vector Shoreline), WDBII (World Data Base II), and DAFIF (Digital Aeronautical Flight Information File). All of these formats are stored as WGS-84 data and relationally organized under CMS. Unlike Project 2851, however, no effort is made to resolve inconsistencies and ambiguities that may exist between the various data types.

4.3.2 Other Standards

In addition to the Distributed Interactive Simulations (DIS) unique data bases specified above, there are a number of other potential sources of simulation data. For the purposes of this discussion, they can be divided into two areas. The first area represents a number of programs specifically oriented to developing standards for various aspects of military simulation. The second area includes military functional communities which regularly use simulations or support them and are consequently involved in collecting data for them. Much of the data from both of these areas could be used to expand the scope of DIS or supplement existing DIS data bases if a common format could be agreed upon.

The list of DIS data bases will continue to grow as increasing numbers of DIS compliant systems come on line and require additional types of data. However, it is hoped that many of these data bases will be standardized by mutual agreement rather than requiring the generation of unique formats and structures solely to support DIS. The following list of potential sources is placed here with the expectation that it will generate inter-community discussions on the availability of data and eventually lead to the adoption of standards and formats useful to multiple communities.

Although it is not specifically addressed here, the DoD Corporate Information Management (CIM) initiative is an important change in the way data bases are developed and maintained. Each Service is currently tasked to develop common data formats for a wide range of uses. Once duplications and ambiguity problems are resolved, the resulting data dictionaries will become standards throughout the Services. It is highly likely that there are other initiatives which DIS is not currently aware of and communities or organizations with which DIS could potentially coordinate to develop common data standards. Any information on such programs would be appreciated.

4.3.2.1 Simulation Standards Programs

In addition to DIS, there are several ongoing programs which are attempting to standardize various aspects of military simulation or simulation data. The following list includes only those programs the DIS community is currently aware of and would benefit from any information readers may have on additional programs which are attempting equivalent or related efforts.

Aggregate Level Simulation Protocol The Aggregate Level Simulation Protocol (ALSP) is a program under the sponsorship of the Defense Advanced Research Projects Agency (DARPA) to develop protocols for linking two or more aggregate models operating in real time as perceived by the human participants. This involves designing a translator which can accommodate significant differentials in the time and space representations used by the various models. In comparison with the second by second event driven updates used in DIS, the aggregated models typically run with from one minute to ten minute time steps. During those time steps, no interrupts from external sources such as another model are allowed. Terrain is also usually aggregated with up to ten or more square kilometers treated as a homogeneous surface, i.e. all forested, all "hilly," all at 1000m altitude, etc. Nevertheless, many of the same functions are performed by both the Aggregate Level Simulation Protocols and the DIS Protocol Data Units (PDUs). Specifically, the primary generators of inter-model messages in both DIS (SIMNET implementation) and the ALSP prototype are the passing of information on platform and unit locations and the notification of weapon launch events.

The aggregate models being used for the ALSP prototype are the USAF Air Warfare Simulation (AWSIM) and the Army Corps Battle Simulation (CBS).

Each of these models is a standard within its respective service. The Mitre Corporation is developing the ALSP protocol translator and coordinating the development of the ALSP protocols. Working with Mitre, Los Alamos Nuclear Laboratories (LANL) is making the necessary modifications to AWSIM and the Jet Propulsion Laboratory (JPL) is making modifications to CBS. Under the present schedule, this prototype is due to be used in Reforger 92. An agreement in principle has been reached to discuss coordination once the ALSP prototype has been successfully delivered. It is anticipated that at some point it would be worthwhile to link DIS to a higher level (aggregate) model. At that point, the compatibility or at least correlation of ALSP and DIS protocols would be of considerable benefit. Consequently, coordination has been opened between DIS and the ALSP program and ALSP has been invited to present their approach and results at a future DIS conference.

Joint Modeling and Simulation System The Joint Modeling and Simulation System (JMASS) is sponsored by the Director of Defense Research and Engineering for Test and Evaluation (DDR&E (T&E)). It is an effort by the intelligence organizations which support the development of threat simulators for the test and evaluation community to produce an object oriented library of reusable Ada software. These objects would represent missiles, aircraft, etc. and the Ada code library would support the efficient expansion of an increasingly more sophisticated, high fidelity air defense environment. The JMASS specification has three levels of fidelity. Only the lowest (dynamic) is expected to run in real time on a workstation sized computer. The level of detail in the emulative version will likely prevent real time implementation. JMASS also plans to generate both two and three dimensional displays of the battlefield on a graphics workstation and to provide Ada code representations of electromagnetic effects. Both of these are of considerable interest to the DIS community.

The JMASS program is being conducted at the USAF Aeronautical Systems Division (ASD) at Wright Patterson Air Force Base. The first module is being built under contract for the Army Missile Intelligence and Space Command (MISC) in Huntsville. The JMASS program held a major symposium in 1991 to present their draft specification to industry and JMASS Working Groups are now being formed in which several members of the DIS community are expected to participate. Currently, the JMASS specification references the DIS standards, however, very little official coordination has occurred.

TRAC Directorate of Data Another organization which is taking a leading role in the standardization of data used in simulations is the TRADOC Research and Analysis Command (TRAC) Directorate of Data Development.This organization is currently supplying data for several Army combat models from a common data base. This electronic data base is drawn from accredited sources such as AMSAA and is regularly updated. The Directorate is also coordinating with the logistics community both to expand its data base and to agree upon common formats to be placed in the Army standard data dictionary being developed under the CIM initiatives.

4.3.2.2. Other Simulation Standardization Efforts

There are a host of military communities and organizations working data standards issues for a variety of purposes. The following list includes only a few and is not intended to imply any preferences. Any additions to the list would be appreciated since they would further the advancement of DIS. This would be both in terms of potentially adding to DIS technical capability and in terms of creating common standards across multiple, usually separated, military communities. For the purpose of consistency, the following organizations are roughly arranged according to the simulation community with which they are most often associated.

Analysis Community The analysis community is the most prolific user of simulations of all types. Most of the combat modeling has been in support of time constrained studies and consequently a great deal of aggregation is usually performed in order to achieve the number of runs needed in the time available to do them. Massive data files have obviously been collected, and in many the question is whether the data is releasable rather than whether the data exists . Because of the great use of models and simulations in this community, the subject of common data bases has been regularly raised. Organizations such as TRAC have actually done something about the problem. Similar programs need to be supported in order to bring about the creation of common data formats and an understanding of what data is actually needed.

Research and Development It has generally been agreed that the Army weapon system data used in DIS implementations will utilize ballistic algorithms from the Ballistic Research Laboratory at Aberdeen Proving Grounds as a standard. This could be expanded to ensure that all weapon effects are in concert with the Joint Munitions Effectiveness Manual (JMEM) data base. The JMEM contains weapons effects data across a wide range of munitions. If it could be agreed that the JMEM (or some subset of it) would constitute one of the DIS standard data bases, several simulation communities could be supported by it. Where extensions are needed for weapons, laydowns, targets, or target sets not addressed in JMEM, agencies such as the Army System Analysis Agency (AMSAA) and its equivalents in the other Services could generate the data. In a related area, the Joint Tactical Coordinating Groups on Survivability and on Munitions have also worked together to provide a wide range of munitions data, damage analysis, and related data. They have then have deposited this data with the Survivability Information Analysis Center (SURVIAC), an industrially funded facility which provides both models and data to the simulation community (primarily air combat and air defense models).

The research and development community operates a number of facilities which potentially constitute sources of data for DIS simulations. Many of these laboratories could both provide data and potentially use access to DIS networks to supplement their research. Typical examples might include the Joint Interoperability Test Center, the Electronic Proving Grounds, and CECOM for information on electromagnetic phenomena. Also, the Army is currently developing a common architecture for the Army Tactical Command and Control System (ATCCS). It is anticipated that such data might very well constitute a rich and certifiable source of information for DIS.

While discussion continues on the proposed use of Project 2851 formats for terrain, both the Defense Mapping Agency and the Army Topographic Engineering Center (formerly ETL) support an increasingly wider range of data bases containing data of potential interest to DIS on features and facilities. Similar large data bases on weather and its effects on visibility, smoke, etc. reside at the Army Atmospheric Sciences Laboratory at White Sands and the Naval Oceanographic Sciences Laboratory. It should be noted that while there are large data bases on the phenomena of weather, there is less data on the effects of weather on vehicle movement, sensor detection (except for visual), firing rates, hit probabilities, etc. Additional information may be available from organizations such as the Army's Center for Lessons Learned (CALL) and similar facilities in the other services.

Test and Evaluation Community Another potential source of data is the operational test community which has the responsibility for conducting realistic tests of battlefield systems. Not only does the community regularly conduct large scale joint operations which constitute a rich source of predictive behavior, but the T&E community is also a major user of simulations especially on larger systems where the ability to test is severely constrained by resources, safety or the availability of realistic threat simulators. Increasingly, the T&E community is involved in modeling and simulation both prior to tests and following tests. For example, prior to the large scale tests of the Mobile Subscriber Equipment, an attempt was made to compare the results of several different communications models to predict test results prior to going to the field. It was noted during coordination meetings that there were few data standards established for characterizing a major communications system above the component level. Consequently the Army Operational Test and Evaluation Command (OPTEC) coordinated the efforts of several government and contractor groups to produce a consolidated set of formats and a data dictionary for the simulation runs. Similar simulation efforts are regularly conducted by the Air Force Operational Test and Evaluation Command (AFOTEC) and both agencies could be both sources of data and participants in the development of DIS standards. OPTEC has already begun investigating the applicability of DIS to their efforts through the NLOS project.

Intelligence Community While the intelligence community does not conduct much simulation, they do provide most of the data on foreign military systems. Much of this is already structured as electronic data bases by the Defense Intelligence Agency (DIA) and the Service intelligence organizations. Systems such as the DIA On-Line System (DIAOLS) and the Community On-line Intelligence Network System (COINS) already contain vast amounts of data in well defined formats. It is probable that some of the intelligence formats could be used directly or that simple translations of those formats could serve DIS needs.

In addition to the standards adopted by the intelligence community, several related initiatives are ongoing. For example, the FORCES electronic data base provides a comprehensive listing of military forces with the added advantage of utilizing a common set of descriptors across multiple types of vehicles built by many nationalities. While much of this data has been developed to support counting rules for treaty verification, it also provides useful definitions to help define concepts such as fidelity. Also, Project 186 for OSD involve comprehensive compilation and subsequent simulation of verified and validated data on U.S., Russian, and third world equipment, forces and, in many cases, doctrine. While this data may not be releasable, (in many cases it is highly classified), it is another potential source of accredited data for applications in the DIS community.

Logistics Community The logistics community was one of the first military communities to make extensive use of computers. The community has subsequently formatted and automated a wide range of logistics data involving maintenance, transportation, and supply. Much of this data is already used in models such as CASMO, TRANSACT, and ASSAULT and thus could constitute a wealth of data. The logistics community is also closely involved with the CIM initiatives and the Army standard data dictionary.

Training Community While it is the training community which will make use of most of the initial DIS compliant systems, the training community has also developed or sponsored many data packages for both existing and future training devices. Probably the most applicable package of training data is the one being prepared for the Close Combat Tactical Trainer (CCTT). While no data format is currently specified, the use of common data formats across dissimilar training devices would promote interoperability and could provide the basis for future DIS standards.

The training community also provides a rich source of data on unit level operations from its instrumented ranges. While access to this data has traditionally been restricted, compilations of this data which did not identify individual units could be very useful in modeling unit behaviors and provide insight into the learning curve factors which should eventually be incorporated into the Semi-automated Forces (SAFOR). Congress required the use of standard protocols in the new MAIS laser engagement instrumentation, and the Army subsequently selected the DIS PDU format to capture this data. By using the DIS format in this manner, data could eventually be analyzed, reformatted and archived for future DIS use without expensive effort. After Action Review formats could also be standardized across many training regimes.

4.4 Interoperability

Interoperability between heterogeneous simulation systems requires that an adequate level of correlation be achieved for the intended application. Correlation in DIS is defined as time and space coherence with respect to the end user, the warfighter. Correlation can be quantified by measuring the degree of coherence at two different points in the data processing stream (see figure 4.4-1): the Entity Local Data Bases and the result of the Entity Processes.

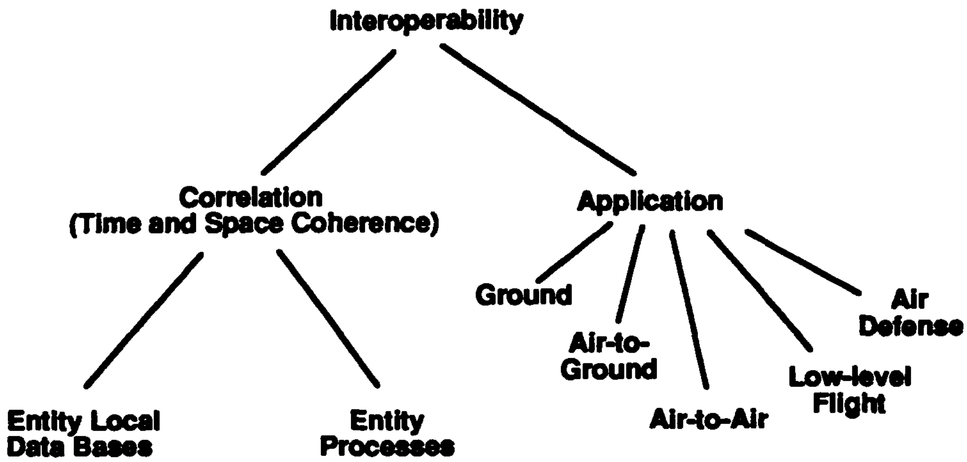


Figure 4.4-1: Interoperability

Entity Local Data Bases are the private data bases associated with the simulation devices; they are derived from public, common source data bases. For example, terrain stored as a polygon mesh is typically proprietary to an IG vendor (the private data base) that derives from DMA or Project 2851 elevation grid data (the public data base). Entity Processes include the rendering algorithms, model positioning functions, and in general all of the processing that is performed by a simulator from the retrieval of the local data base data to the output of the processed scene. Figure 4.2.1-1 illustrated how DIS message and data base standards are used to manage and control simulation entity local data bases and processes.

Correlation Metrics

Time and space coherence can be represented notionally as a two dimensional correlation space, where the two axes are defined by time and space fidelity vectors for a given application or exercise. Each axis can in turn be viewed as a composite expression of the numerous factors that affect time and space coherence - time coherence by factors such as update rates, latencies, vehicle dynamics and network bandwidth, and space coherence by factors such as location, attitude, geometry, and appearance. See Figure 4.4-2.

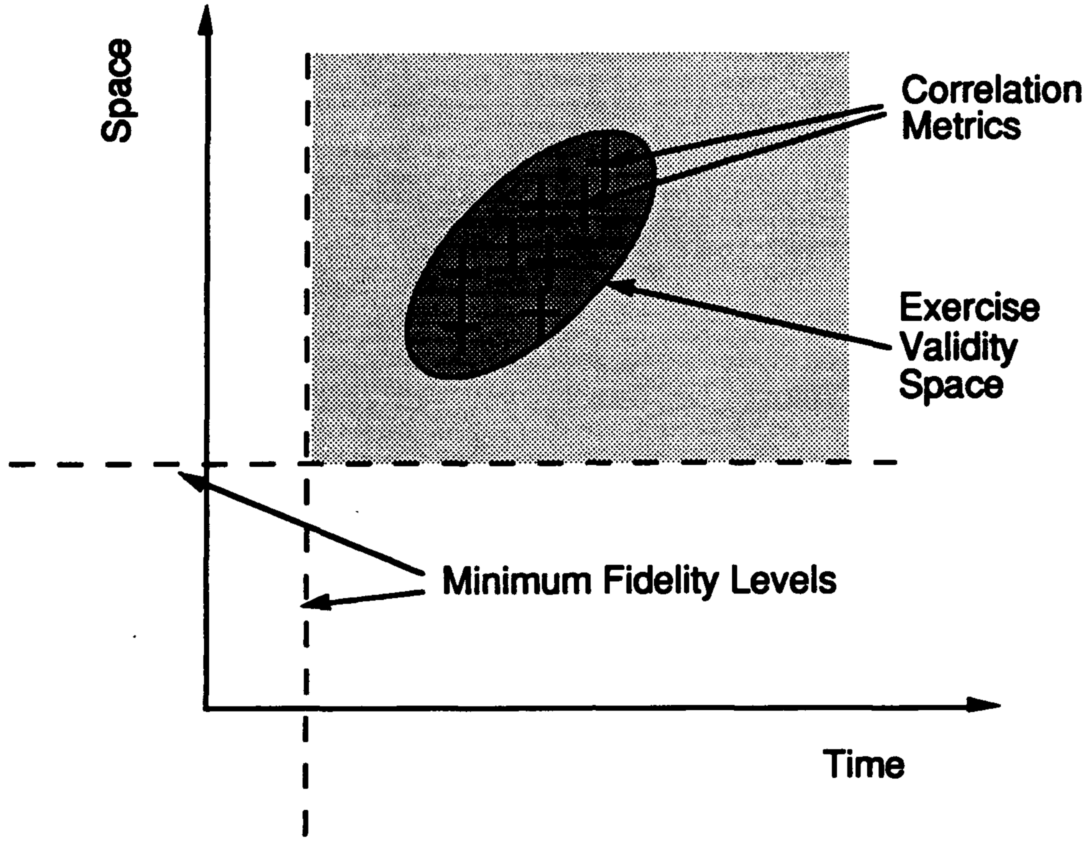


Figure 4.4-2: Correlation Space

Within this correlation plane, we define a minimum level of fidelity on each axis that must be achieved for a given exercise (shown shaded in Figure 4.4-2). The simulation entity correlation metrics are then plotted in the time/space plane, and tested to determine whether or not they fall within an "Exercise Validity Space", shown notionally in Figure 4.4-2 as an ellipse. The notion is that the various entities that make up an exercise must lie relatively close together in the time/space plane for a valid exercise to take place. The location of the ellipse boundary will ultimately be determined by warfighter-in-the-loop experiments to test the correlation hypotheses for each application.

This correlation construct introduces the concepts of relative and absolute correlation. Relative correlation is defined as "closeness" in the time/space plane, the tendency of the metrics to cluster. Relative correlation says something about the similarity between two or more simulation entities, whereas absolute correlation measures the simulation entity against a fixed reference, such as an external source data base. In this sense absolute correlation describes the fidelity of a simulation entity. Clearly it is necessary to consider both aspects of correlation to determine interoperability for a given exercise.

5.0 Network Issues

This section describes the networking considerations of the DIS architecture. The architecture provides a robust capability to establish virtual networks maximizing the use of commercial products and services based on open, non-proprietary protocols and minimizing the development of items unique to DIS. The virtual networks can be sized to meet DIS needs for numbers of simulation entities, throughput, latency, and security in a cost effective manner. The use of products and services based on open standards allows for the interoperability of heterogeneous Local Area Networks (LANs), Wide Area Networks (WANs), and interworking technologies, i.e., bridges, routers, and gateways. Interoperability of simulators having different levels of fidelity is achieved by Communications Interface Units (CIUs) and Communications Adapter Units (CAUs) that operate at the level of the DIS PDUs and use standard protocols for communications. Commercially developed security technologies driven by the National Security Agency (NSA) allow the implementation of classified DIS exercises at the Secret level with provision for Special Access Programs (SAP).

In contrast, SIMNET was designed to demonstrate the technical feasibility of networked simulators with a homogeneous environment of Ethernet LANs, Ethernet bridges, simulators having the same fidelity and unclassified exercises. The SIMNET design stressed the performance envelope of the networking technology available at the time. The SIMNET LAN protocol was specially designed to meet throughput and latency needs that could not be met by the use of the Internet Protocol (IP) on the selected hardware. This special purpose protocol is not a commercial standard. Also, the STream (ST) protocol implemented in the Terrestrial Wideband network (TWBnet) used to demonstrate a SIMNET WAN capability is not a commercial standard. The ST protocol provides bandwidth reservation and multicast capability. The existing TWBnet technology is proprietary even though the standard specification and software code are in the public domain.

Processing and communications technologies have advanced significantly beyond the technologies available to the SIMNET developers. Processing technologies can now support DIS needs for communications throughputs and latencies using standard protocols implemented in commercial products and services. In particular, WAN technologies offer sufficient bandwidth reservation using virtual circuits and the ability to burst above the reserved bandwidth to the capacity of the physical transmission facility. Commercial multicast standards are before national and international standards bodies for ratification and products are emerging.

The DIS architecture provides a structure by which independently developed systems may interact with each other using virtual networks implemented with widely available commercial products and services. The virtual networks are configurable to meet DIS throughput and latency needs. The 1.5 Mbps T-1 networking technology can support concurrent exercises of 1,000 simulation entities distributed over several sites. Sites may need two or three T-1 lines to accommodate incoming DIS PDU traffic. Early 1990s technology at 45 Mbps T-3 rates will support concurrent exercises of 10,000 simulation entities while next generation Switched Multimegabit Digital Services (SMDS) using Synchronous Optical Network (SONET) technologies of 50 Mbps to 2.4 Gbps will support concurrent exercises of 100,000 simulation entities by the mid 1990s. Communications services defined by the virtual network are separable from the other elements of the DIS architecture and thus can be implemented with widely available COTS (Commercial-Off-The-Shelf) products and services. The architecture documents needed extensions to communications standards applicable to DIS. The primary extension is the standardization of multicast services and protocols in the mid 1990s. The architecture thereby minimizes the effort required to develop network configuration items unique to DIS.

5.1 Conformance to BDS-D Technology Development Plan (TDP)

This section addresses the flowdown and derivation of DIS Architecture constraints. The Virtual Network Architecture is compliant with the following: 1) DIS reference model, 2) Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation, Military Standard (Draft), IST-PD-90-2 (revised), and 3) BDS-D ATTD Technology Development Plan (TDP). The virtual network architecture is interoperable with virtual networks conforming to 1) ISO OSI Reference Model , 2) the SIMNET embedded base, and 3) Communications Architecture for DIS (CADIS) (Draft), Communications Architecture and Security Subgroup (CASS).

The current SIMNET baseline and exit criteria for networking from the Phase I DIS architecture of the TDP is shown below.

OPERATIONAL CAPABILITY	CURRENT BASELINE	EXIT CRITERIA	ENHANCED GOAL
Local Area Networks (LANs)	Identical	Mixed	
LAN Inter-Connect	Bridges	Wide Area Network (WAN)	Intelligent WAN Gateway
Fidelity	Same	Mixed	
Classification	Unclassified	Secret	SAP

Table 5.1-1. DIS Phase 1 Architecture Networking Objectives.

The networking architecture supports all the exit criteria and enhanced goals in Table 5.1-1. Additional networking objectives derived from the TDP are as follows:

- *Leverage existing and emerging communications standards and protocols; work to extend standards and protocols in standards organizations to address deficiencies.*

- *Support adaptation of networking resources to meet program needs for reconfigurable DIS.*
- *Support trade-off of network costs with simulator costs within the overall framework of DIS costs.*
- *Optimize network costs to meet simulation preparation, SESSION scheduling, virtual network connectivity, throughput, and delay criteria.*

The TDP also calls attention to the following enabling technologies that can provide the added capabilities: 1) computer networking protocols and standards to link dissimilar simulations and simulators, and 2) local, wide area, and long haul networking technologies.

DoD policy mandates the use of the Government Open Systems Interconnection Profile (GOSIP) for communications between computers. DoD has stated the intent to migrate to GOSIP from the embedded base of non-GOSIP compliant systems. The National Institute of Standards and Technology (NIST) has the responsibility within the federal government to validate and certify commercial products for compliance with GOSIP.

The CADIS document specifies latency requirements based on human perception of computer image generation and networked voice communications simulated by DIS messages. The simulator end-to-end latency for real-time data and voice messages should not exceed 300 [To Be Resolved] msec. The allocation to the virtual networking component of the architecture results in a latency of <150 msec [TBR] for 95% of the DIS PDUs for real-time data and voice messages.

5.2 Two-Tiered Virtual Network Hierarchy

The DIS reference model uses a two tier network structure based on the concept of a virtual network constituting the means for message communication within a cell and a Communications Interface Unit (CIU) or Communications Adapter Unit (CAU) providing the interworking across cells. The two-tiered hierarchy is illustrated in Figure 5.2-1.

Sections 5.3, 5.4, and 5.5 document the Virtual Network, CIU, and CAU reference models. The reference models identify the functions to be performed by the Virtual Network, CIU, and CAU without imposing performance constraints or determining where and how the functions are implemented. The architecture is validated against the reference models as well as program and performance constraints.

A cell is defined for the time duration of a SESSION to be a homogeneous collection of simulation entities sharing one set of common cell databases supported by a virtual network. By definition, all state messages are broadcast within the cell. To satisfy the homogeneous criteria, the simulation entities are required to have consistent behaviors.

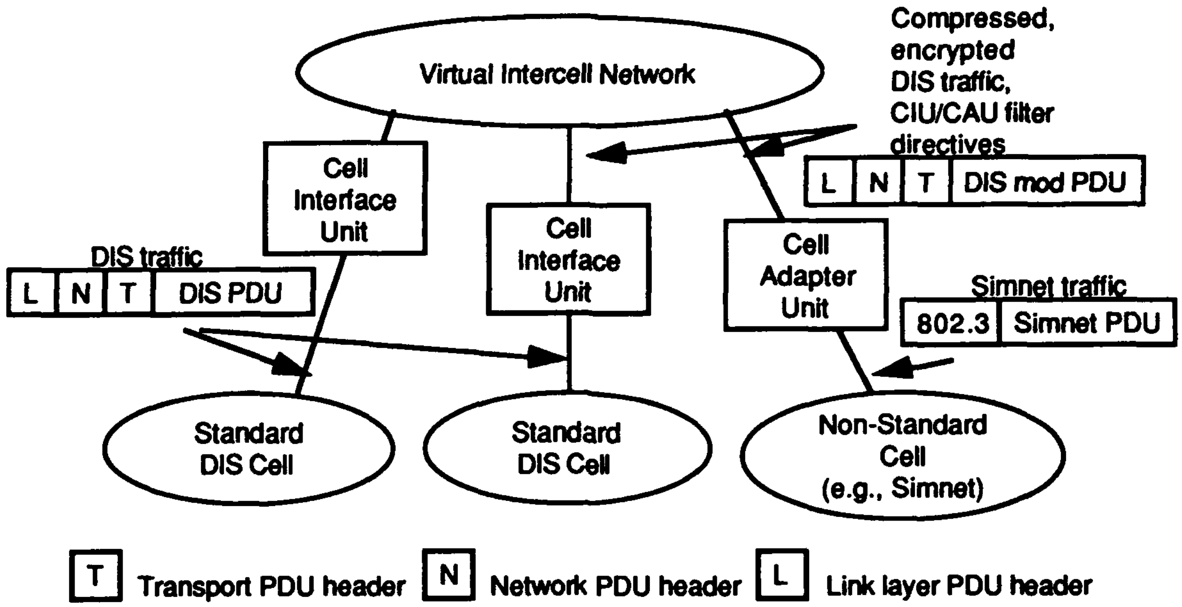


Figure 5.2-1. Two-tiered network hierarchy.

A cell can be composed of resources at a single site or span multiple sites. A site can concurrently contain multiple cells. The virtual network within a cell can consist of any combination of LANs and WANs interworked with bridges, routers, and gateways. Multiple cells at a site can share a physical communication network that is partitioned by the CIU or CAU into multiple virtual networks.

Message blocking based only on message addressing is a network function, rather than a CIU function. The network is responsible for establishment of the virtual network and for multicast/multipeer communication.

Simulation entities communicate via a virtual network. Message communication to other cells is via the CIU or CAU.

Simulation entities communicate the following types of messages via the virtual network: 1) entity state, 2) entity interaction, 3) management, 4) environment, and 5) voice communication. The reference model does not address video teleconferencing communication. (Video teleconferencing communication in support of, but not part of DIS is available from service providers. See the Network Topology discussion in Volume 2 of this document.)

5.3 Virtual Network Reference Model

The Virtual Network reference model is shown in Figure 5.3-1 in relationship to its interfaces with the CIU/CAU, simulation entities, simulation support entities, and databases. The components of the virtual network are 1) Local Area Networks (LANs), 2) access to Points of Presence (POPs), 3) Wide Area Network (WAN), also called Long Haul Network (LHN), 4) interworking, and 5) network management.

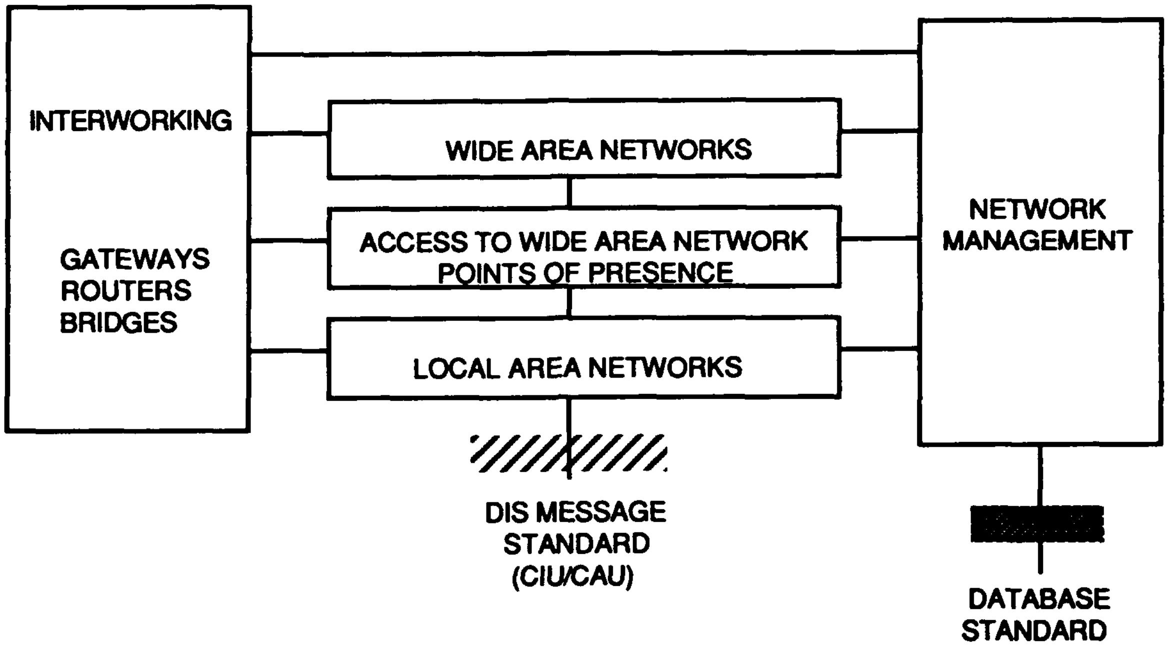


Figure 5.3-1: Virtual network reference model and standards.

5.3.1 LAN Reference Model

The LAN reference model supports hundreds to thousands of simulation entities per LAN. DIS and non-DIS entities may share a common LAN at the expense of a CAU and translation of PDUs with the additional overhead of doubling the occupancy of the LAN.

LAN maximum data rates are shown in Table 5.3-1.

Table 5.3-1. LAN maximum data rates

LAN	Maximum Rate
Token Ring (Copper)	4/16 Mbps
Ethernet	10 Mbps
FDDI and FDDI II	100 Mbps
SONET	51.84 Mbps to 2.488 Gbps

The LAN reference model uses existing and emerging technologies based on a common broadcast transmission media connecting all the simulation entities. This media may be based on twisted pair copper wires, coax cable, or fiber optic cable. Electrical cables can be shielded to prevent compromising emissions to meet security policies for secret and Special Access Program (SAP) levels of classification.

Voice traffic carried on simulated Combat Net Radio (CNR) can be integrated on IEEE 802.3 (Ethernet), IEEE 802.4 (Token Bus), IEEE 802.5 (Token Ring), FDDI, FDDI II, and IEEE 802.6 (SONET) LANs. However, the Ethernet can carry only a limited number of simulated CNR channels whereas there are few restrictions on the other types of LANs. Simulated CNR can be circuit switched at a site if the site communications system supports an adequate number of digital conference bridges.

5.3.2 POP Access Reference Model

Cells requiring a WAN connection access the WAN through a Point of Presence (POP). Transmission lines connecting the site to the POP may be provided by the Local Exchange Carrier (LEC) or by using microwave radio or fiber optic technology to bypass the LEC. Interexchange Carrier (IXC) facilities may be required to connect to a WAN service point if the POP for that service is located in a Local Access and Transport Area (LATA) different from the site's LATA.

The data rates that are generally available for connecting a site to a POP are shown in Table 5.3-2.

Table 5.3-2 Data rates available for connection to a POP

Facility Type	Data Rate
DS-0	56 kbps
DS-0 A	64 kbps
Fractional T-1	N x 64 kbps (N < 24)
ISDN Basic Rate Interface (BRI)	144 kbps
ISDN Primary Rate Interface (PRI)	1.544 Mbps (1.536 Mbps usable)
T-1 (DS-1)	1.544 Mbps (1.536 Mbps usable)
E-1, E-2, E-3, E-4 European Standards (CEPT)	2.048 Mbps (1.920 Mbps usable) 8.448 Mbps (7.68 Mbps usable) 34.368 Mbps (30.072 Mbps usable, 139.264 (122.88 Mbps usable)
T-3 (DS-3)	44.736 Mbps (43.008 Mbps usable)
SONET (IEEE 802.6) - OC1, OC3, OC9, OC12, OC18, OC24, OC36, OC48	51.84, 155.52, 466.56, 622.08, 933.12 Mbps; 1.244, 1.866, 2.488 Gbps

Access facilities from a site to a POP may or may not be integrated (shared) with other communications services at the site.

5.3.3 WAN Reference Model

The WAN reference model is shown in Figure 5.3-2 in relationship to its interfaces with the POP access, interworking, and management components.

The elements of the WAN are: 1) physical signals (electrical, optical, radio frequency, infrared), 2) signal format, 3) multiplexing/demultiplexing, 4) reliability, 5) grooming, 6) switching, 7) transmission, and 8) monitoring and control. Different classes of communications, e.g., voice and data, may transparently use separate WANs or be integrated on a common WAN. The interworking of WANs is transparent to DIS given that latency, throughput, and security constraints are not violated.

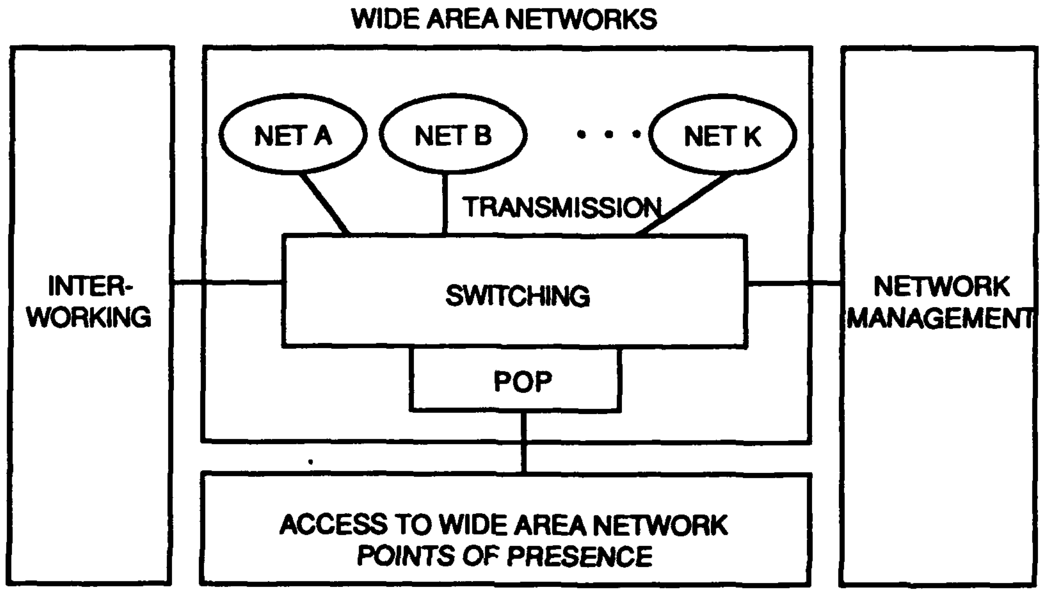


Figure 5.3-2: Wide area network reference model.

5.3.4 Virtual Network Interworking Reference Model

The Virtual Network Interworking reference model is shown in Figure 5.3-3 in relationship to its interfaces with the virtual network transmission and management components. The elements of virtual network interworking are: 1) signal translation, 2) protocol translation, 3) rate adaptation, 4) reliability, and 5) monitoring and control. Interworking is accomplished by bridges, routers, and application gateways at the link, network, and application layers, respectively, of the OSI reference model.

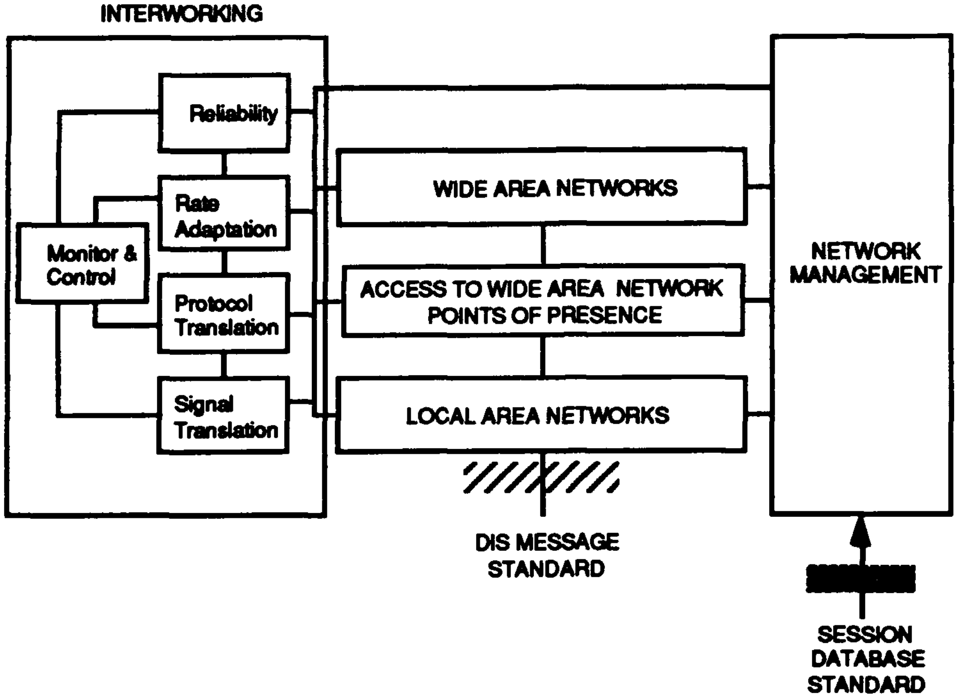


Figure 5.3-3: Virtual networks interworking reference model and standards.

5.3.5 Virtual Network Management Reference Model

The virtual network management reference model is shown in Figure 5.3-4 in relationship to its interfaces with the LAN, POPs access, WAN, and interworking components. The elements of virtual network management are: 1) fault management, 2) configuration management, 3) accounting management, 4) performance management, and 5) security management.

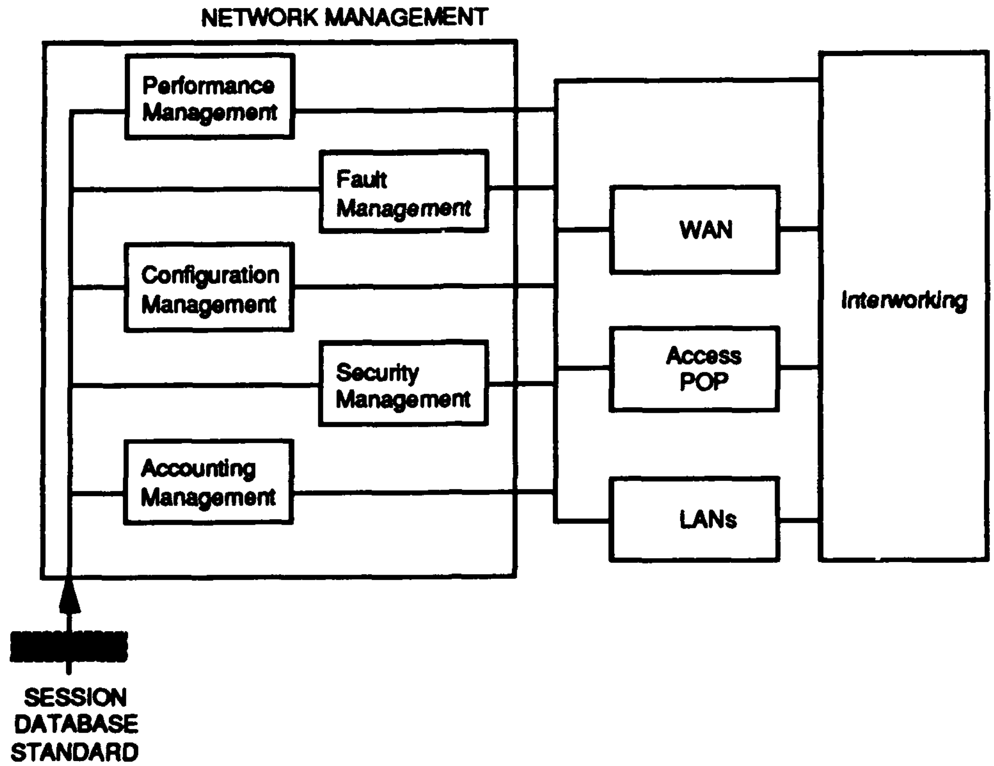


Figure 5.3-4: Virtual network management reference model and standards

Fault management is the detection and monitoring of abnormal network operations. Faults manifest themselves as particular events (errors) in the operation of the network. Fault management uses the set of facilities used to manage error logs, accept and act on error notifications, trace faults and carry out sequences of diagnostic tests.

Configuration management provides for the identification and control of managed objects with the goal of insuring the continuous operation of communications services. Configuration management uses the set of facilities required to initialize and close down managed objects, to collect the data necessary to determine the system's configuration state, to change the configuration of the system (switch in standby equipment) and to associate logical names with sets of managed objects.

Accounting management is the determination of costs for the use of managed objects and the establishment of charges for this use. Accounting management uses those facilities which inform users of incurred costs, enable the fixing of account limits for the allocation of resources and provide for the combination of costs when multiple managed objects are invoked to accomplish a communications task.

Performance management is the evaluation of the long term behavior of managed objects. This differs from fault management or configuration management in that both of these tend to focus on the immediate status of a managed object such as, "Is it on?", or "Is a standby available? " The information used in performance management is typically statistical data that is analyzed to determine and predict trends in the communications capabilities of the network.

Security Management uses the facilities required to implement an organization's security policy as it applies to the communications aspects of a network. Specific functions included under security management are the control and maintenance of access restrictions, the management of encryption keys and the creation and distribution of security logs, such as access audits.

A management task may require the use of services provided for under multiple specific management functions. For example, "It's broke, fix it!" would require fault determination and isolation (using fault management) and systems reconfiguration, such as changing routing tables or switching in standby equipment (using configuration management).

5.4 Virtual Network Standards

Figure 5.4-1 shows the DIS virtual network standards for LANs, POPs access, WANs, interworking, and network management. The standards conform to federal government policy to migrate communications networks to OSI. These standards are commercially developed and certified by NIST.

5.4.1 LAN Standards

LAN communications use the IEEE 802 series standards for physical and data link layers and the OSI CLNS/CLNP for network and transport layers to support the communication of DIS PDUs. Bandwidth reservation is not guaranteed for 10 Mbps Ethernet networks using a CSMA/CD scheme. Such LANs are required to have their traffic limited to 6 Mbps to reduce contention to a level that preserves the time-space coherence of an exercise.

Intra-LAN multicast service is based on the use of the PDU exercise ID, simulation support entity administration and the enforcement of a multilevel security policy for classified SAP exercises.

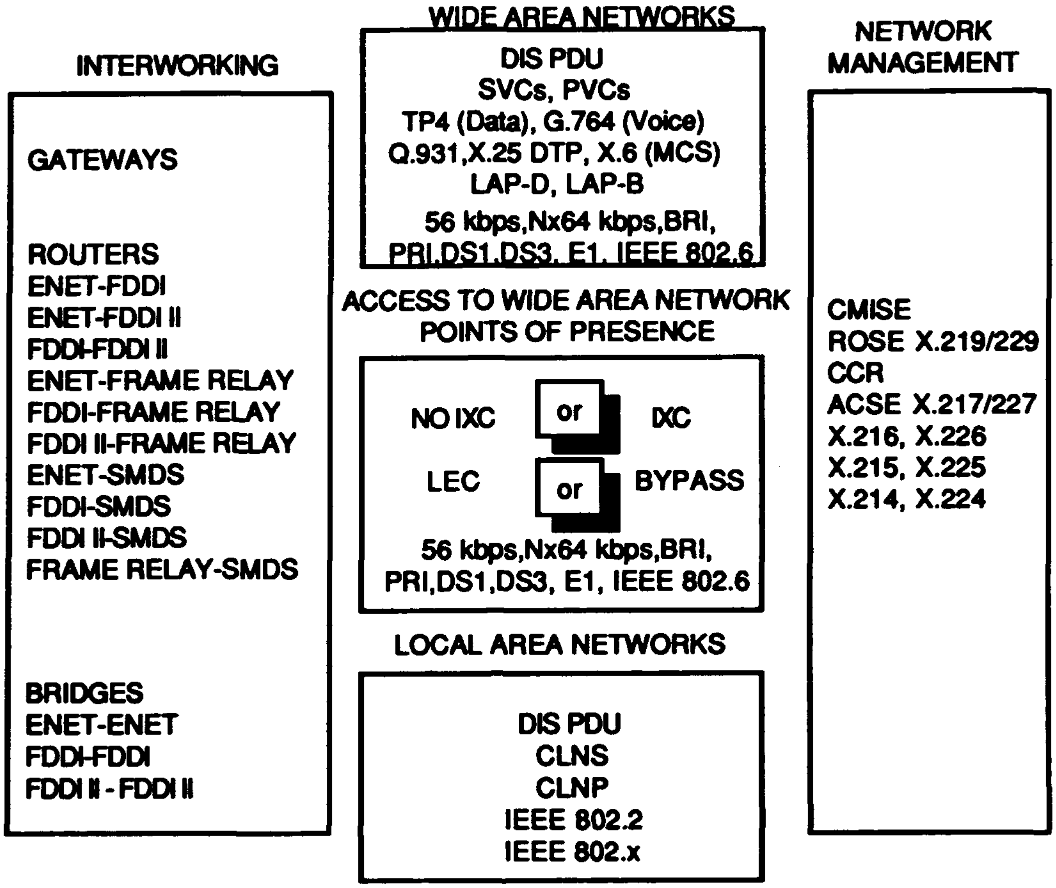


Figure 5.4-1: DIS virtual networking standards.

5.4.2 WAN Standards

Note that the LANs use the OSI Connectionless Network Service (CLNS) and Protocol (CLNP), whereas the WANs use a connection oriented channel via Transport (TP), X.25 Data Transfer Phase, and Link Access Protocol (LAP-B or LAP-D) provided by Frame Relay or SMDS services. Bridges perform the interworking between the connectionless form at the LAN and the connection form at the WAN. The connection oriented structure of the WANs uses either Switched or Permanent Virtual Circuits (SVCs, PVCs) to provide an average bandwidth reservation capability between sites to support real-time voice and data communications. The voice communications use a Packet Voice Networking Protocol (PVP) based on CCITT Standard G.764. The SVCs and PVCs are connection oriented only in the sense of connecting sites and are not connection oriented from the perspective of a simulation entity. The SVCs and PVCs are in essence extensions of the broadcast media of the LAN to interconnect LANs at other sites. SVCs and PVCs support traffic bursts above the reserved data rate constrained only by contention from other SVCs and PVCs sharing the transmission facility and the maximum usable rate of the transmission facility.

The draft X.6 Multicast Communications Service (MCS) and Protocol (MCP) provides One-Way, Two-Way, and N-Way multicasting among sites. DIS uses the N-Way capability. X.6 is based on a connection oriented paradigm and is not intended for connectionless traffic as would occur on a LAN. The multicasting service may be provided on the edges of a WAN in the gateways, routers, and bridges. Alternatively, the multicasting may be embedded within the WANs to avoid n replications of a DIS PDU to save transmission bandwidth and expense from a site to the POPs.

Multicasting on a LAN can be implemented in the Layer 2 Logical Link Control (LLC). For example, the DIS PDU exercise field can be mapped to a LLC address on the link level PDU. Preliminary work has begun to define a connectionless multicast protocol for the network layer in OSI but has not reached the same stage as X.6 as described above.

5.4.3 Network Management Standards

ISO standards committees employ an abstract model, the System Management Model, to organize the services offered by an OSI compliant network management system. Specific services and protocols are defined in related protocol specification standards. Several extensions have been incorporated into the basic model. These extensions provide additional functions used to facilitate information transfer within a large network. One of these is the OSI Network Management extension. The OSI Management Environment is defined as "that subset of the total OSI Environment which is concerned with the tools and services needed to control and supervise interconnection activities and managed objects". A managed object could be a piece of hardware, a software component or a collection of information such as a database. The object does not need to be an OSI resource, as it can fall outside of the framework established by the Reference Model. For example, a Protocol Data Unit (PDU) Manager for DIS can be defined as a managed object outside the framework of the OSI model.

ISO management standards address both the syntax and semantics of the information required to accomplish the resource management. They also specify the communications services required to transport this information within an OSI environment. The standards do not specify how specific management functions are accomplished. That definition falls under the domain of the user application programs.

Figure 5.4-2 depicts the organization of resource management within the OSI environment. OSI management focuses on the monitoring and control of "managed objects," where a managed object can be any resource (hardware or software).

As shown in Figure 5.4-2, the systems management applications and the user's interaction with these applications fall outside of the scope of OSI management standards. Three forms of management information exchange are defined within the OSI management architecture. These are:

- Systems Management
- Layer Management
- Layer Operation

This set of standards is given by ISO/IEC 7498-4, *OSI: Information Processing Systems - Open Systems Interconnection - OSI Management Framework*. This standard, an extension to the original OSI Reference Model, introduces the concepts of systems management, layer management and communications protocol management functions.

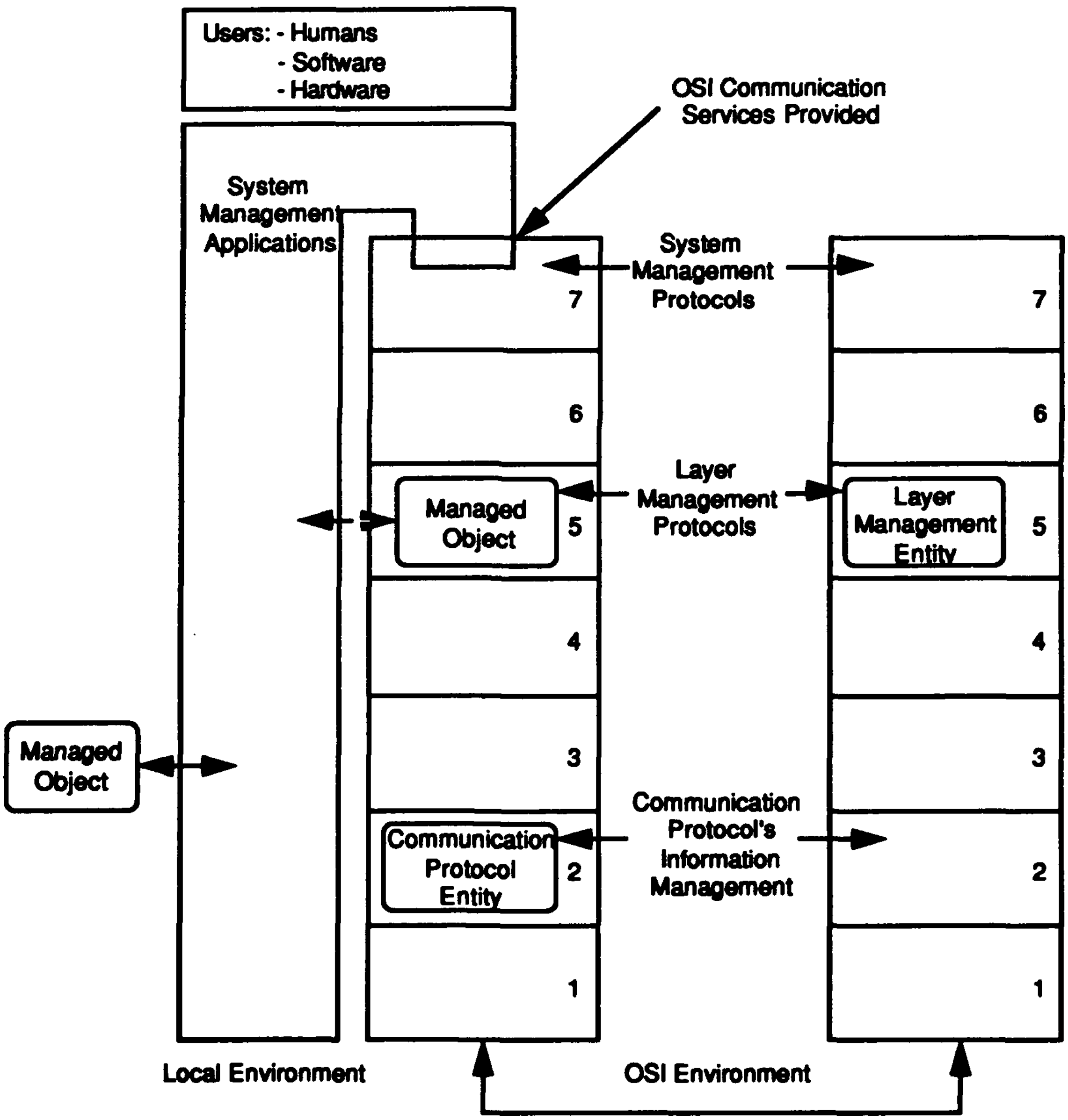


Figure 5.4-2: OSI Management.

Systems management is the preferred form of management information exchange. Systems management provides mechanisms for the monitoring and control of all managed objects. Systems management is the only means by which OSI management of multiple layers is accomplished.

The systems management standards contain seven items. Each item defines a particular management function, such as an object management function, a confidence and diagnostic testing function and an error reporting and information retrieval function. Some of these items reference the actual messages that are to be employed in communicating the information required to invoke the function. The management functions are grouped into the Specific Management Functional Areas (SMFAs), such as fault and configuration management.

Layer management provides for the monitoring and control of managed objects within a given layer. Layer management protocols should only be used when either the systems management services do not support the exchange of layer management information or when the exchange is not supported by higher layer services. Layer management entities are processes, which are separate from those used to provide the communications functions. As such, layer managers can maintain logs containing parameter values related to specific communications functions such as average delays between entities. In addition layer managers can test the services provided by the layer beneath them.

The *management information* standards describe the organization of information used by OSI management applications. One describes the structure of managed objects, including the concept of object attributes and the process used to assign a name to classes of managed objects. A second defines a group of object attribute types that may be applicable to most managed objects. Included as part of the attribute definition is the Abstract Syntax Notation (ASN.1) definition of the attribute. This is an ISO standard structure used to encode information related to the attribute. The third defines a number of object classes (groupings of managed objects) that may be used as "superior" classes when defining new classes of managed objects. The fourth tells how to use the first three items in this set.

Management functions within the communications protocols themselves are referred to as layer operations. They differ from layer management functions in that as soon as that instance of the protocol is not needed the layer operations no longer exist. Examples of information conveyed within the communications protocols are:

- Error information for that particular instance of communications.
- Parameters used to modify the protocol during that instance of communications.
- Parameters used to control the establishment or release of a specific connection.

There is only one management protocol specifically designed to exchange systems management information. This protocol, the Common Management Information Protocol (CMIP), is used by the Common Management Information Service Element (CMISE). CMIP is defined in two standards. ISO 9595 defines the services, i.e., CMIS, used to exchange systems management information while ISO 9596 specifies the actual protocol used to provide these services. The CMIP standard specifies the use of services provided by another protocol, the Remote Operation Service Element (ROSE) protocol.

CMIS/CMIP addresses acknowledged limitations of the Simple Network Management Protocol (SNMP) to communicate status of large numbers of managed objects in a structured, efficient, timely, reliable manner.

5.5 Virtual Network Latency and Throughput

Communications systems are to be engineered on the basis of peak (i.e., short time interval) average PDU rates three times the average (i.e., long term over the duration of an exercise) PDU rate for simulation entities participating in an exercise.

Table 5.5-1 provides some examples of the latencies that are available today. We believe that this is the level of network performance that will be utilized by DIS applications for the next 5 to 10 years.

Table 5.5-1: DIS PDU networking latency allocation

Network Component	Average Latency	95% Percentile Latency
Originating Simulation Entity LAN Interface	7 msec	20 msec
Originating WAN Interworking Interface	7 msec	20 msec
WAN	22 msec	70 msec
Receiving WAN Interworking Interface	7 msec	20 msec
Receiving Simulation Entity LAN Interface	7 msec	20 msec
Total LAN-WAN-LAN Latency	50 msec	150 msec

Information specifying the required network latency characteristics for a simulation environment would be defined in the SIMWORLD portion of an Exercise Database. The SESSION portion would be used to select the capabilities of the actual networks used in the exercise. The two portions would then be compared to determine the ability of the specific SESSION configuration to support the application.

WAN communications channels will be engineered for 80% occupancy at peak average DIS PDU rates.

LANs are be to engineered as follows:

- CSMA/CD: 60% occupancy at peak average DIS PDU rates
- Token: 80% occupancy at peak average DIS PDU rates

Processing is to be engineered for 80% occupancy at peak average DIS PDU rates.

The communications technologies required to support the envisioned numbers of simulation entities are shown in Table 5.5-2.

Table 5.5-2: Communications technologies to support numbers of simulation entities.

Number of Simulation Entities	LAN/WAN Communications Technology	LAN/WAN Communications Data Rate
1,000	Ethernet/T-1	10 Mbps/1.544 Mbps (2 - 3 T-1s per site)
10,000	FDDI/T-3	100 Mbps/44.736 Mbps
100,000	SONET (OC3 to OC12)	155.52-622.08 Mbps

6 DoD Policy on Modeling and Simulation

The Department of Defense (DoD) has undertaken a major effort to promote the effective and efficient use of modeling and simulation (M&S) in joint education and training, research and development, test and evaluation, logistics and readiness, and operations and cost analysis. To support this initiative, the Deputy Secretary of Defense has issued a policy letter directing that the Under Secretary of Defense for Acquisition (USD (A)) is responsible for strengthening the use of modeling and simulation throughout the DoD. This same policy letter established the DoD Executive Council for Models & Simulations (EXCIMS) and the Defense Modeling and Simulation Office (DMSO) to advise and support the USD (A) in the execution of these responsibilities.

The charter of the EXCIMS and DMSO includes:

- Establishing OSD cognizance and facilitating coordination among DoD M&S activities.
- Promoting the use of interoperability standards and protocols where appropriate.
- Promoting the use of common data bases.
- Stimulating joint use, high return M&S investment such as reusable code.

The DOD policy states that maximum use will be made of accepted professional and commercial practices and existing DoD and Component programs and procedures. The standards recommended by the DMSO and approved by the EXCIMS will generally apply to those future models and simulations specifically designated for use at the Department level in support of joint education and training, research and development, test and evaluation, and operation and cost analysis in support of Joint Required Operational Capability (JROC), Defense Planning Resources Board (DPRB), or Defense Acquisition Board (DAB) deliberations. Because of the likelihood that almost all future models and simulations will involve joint operations either directly or through the linking of separate Service models, it is anticipated that this policy will apply to most future M&S.

6.1 Applicability of DoD M&S Policy to the DIS Architecture

The DIS architecture document addresses several OSD policy concerns. Foremost of these, the DIS architecture is designed to maximize interoperability among dissimilar simulators. In line with the DoD policy, the architecture also recommends that plans for configuration management, verification, validation, accreditation, and releasability of DIS compliant M&S be developed. As specified in the architecture, all standards and related information needed to participate in DIS are to be open and available to Government, industry, and academia.

Through semiannual symposiums and related workshops, the DIS Steering Committee and DIS Working Groups promote M&S interoperability and reduce duplicative developments by providing a forum for discussion and demonstration of advanced technologies in support of vehicle and environmental simulation and related areas.

The DIS architecture document is not a policy statement. Consequently it does not attempt to set policy. On the other hand, in developing the architecture, it was regularly noted that certain problems could only be adequately addressed in policy. The following paragraphs address those areas in order to identify them for future action by applicable policy makers.

6.2. Policy Issues related to DIS Standards

Having standards for message protocols and data bases is a necessary, but not sufficient condition for DIS compliance. There must also be a means of determining conformance with those standards, correlating heterogeneous systems which each claim compliance with the standards, and benchmarking the levels of compliance. The DIS architecture supports the concept of rapid determination of DIS compliance for specific simulation entities. This validation process is primarily based on inheritance of validity through the use of specific standards, models, data bases, procedures, etc. which have been previously shown to produce the desired results. However, this inheritance cannot be assumed unless it can be shown to be identical or traced back to a previously compliant implementation.

6.2.1 Conformance

The responsibility for verifying and validating DIS implementations is currently left to the same user groups which would likely be assigned the responsibility for accrediting a networked simulation for a specific exercise, test, or study. This is not a good solution since it potentially generates conflicts of interest or at least the perception of conflict of interest. As the DIS process matures, it is expected that a certification process will be created to determine compliance with the DIS PDU standard (soon to be voted upon by the IEEE and other standards bodies). As other DIS standards are developed, they would also fall under this certification process. An existing organization could be assigned or awarded the responsibility for determining compliance, or a new group (potentially supported by user fees) could be created to conduct it. The purpose of the compliance group would be to independently verify and to some extent validate new DIS compliant data, data formats, models, algorithms, etc. and to establish the validity of specific implementations. A potential model for this certification process is the Software Engineering Institute (SEI) methodology for determining a facility's level of software development capability on a scale of one to five. This program is essentially industrially funded except for a relatively small staff at SEI and constitutes an excellent prototype for a similar DIS facility.

6.2.2 Correlation

It is possible to have standards, conform to them, and still not achieve either interoperability or realism to the degree desired. The DIS architecture accommodates heterogeneous simulators (built by different manufacturers using different proprietary techniques). Consequently, it is necessary to provide some way to determine that these DIS compliant simulators (with dissimilar internal representations made compliant through the use of common protocols or translators) correlate to the desired representation of the real world for a given session (selected fidelity). It is also necessary to determine that two or more such dissimilar simulators can interoperate effectively within a given range of fidelity (fair fight). At the present time, there is no way to determine this without conducting a "test drive" with a cross section of experienced crews. However, the DIS architecture supports the later development of correlation techniques and factors. It is anticipated that at least some quantitative criteria can be determined which would eventually become benchmarks against which comparisons of both levels of fidelity and degrees of interoperability can be made.

6.2.3 Benchmarks

Part of the standardization and potentially part of the certification taking place under the DIS process will eventually include designated benchmarks to represent varying levels of fidelity. At the most detailed level, there will be specific results which an algorithm should match within some given margin of error if it is to be considered compliant with the DIS standard. This is most likely to occur first with different implementations of the remote entity approximation (dead reckoning) algorithms, but it could also apply to movement algorithms, propagation equations, ballistics and similar quantitatively specifiable items throughout the DIS architecture. Benchmarks could also be applied to computer image generators with respect to characterizing minimum scene management capabilities without necessarily specifying what implementation technique is applicable. Benchmarks are also applicable to the communications networks supporting DIS, where they can be used to test against latency standards under a given set of conditions.

On a less quantitative level, there also exists the opportunity to benchmark certain qualitative elements within the architecture. These benchmarks would help determine the bias introduced into the exercise by the simulator. This might include knowledgeable user comparisons with "visual standards" which could include renderings of screen displays, photographs of vehicle control panels, etc. Since many of these "standards" involve human evaluation of qualities such as realism, resolution, etc, the results will likely be somewhat fuzzy. However, we should be able to rank order visual systems against a relative, but visually defined standard for specific applications. This should at least allow categorization of systems or system components into definable groupings and allow the community to depart from virtually meaningless terms such as high or low fidelity.

At the most abstract level, an entire exercise may be benchmarked against the outcome of a specific battle, e.g. the Battle of 73 Easting from Desert Storm, another historical battle, or specific instrumented exercises such as those conducted at the National Training Center or the Joint Readiness Training Center. Since the underlying concept of DIS is the ability to conduct free play exercises, the unit level or battle level benchmarks would be loosely defined. Nevertheless, to foster credibility, DIS exercises should be able to recreate benchmarked scenarios with outcomes (movement rates, casualties, consumption rates, etc.) within reasonable bounds of historical accuracy. Since any simulation can be "tuned" to replicate a specific battle, the acid test is the ability to replicate several battles making only those data changes which reflect identifiable differences in the battles.

6.2.4 DIS Relationships to Other Standards

DIS is not officially designated by DMSO or the DoD EXCIMS as providing standards meeting DoD M&S policy, but the Army has designated DIS as the standard for its family of Combined Arms Tactical Trainers (CATT) and for incorporation into its next generation of laser tactical engagement instrumentation. Likewise, the Navy has designated DIS standards be used on their next generation Tactical Control Training System. While distributed interactive simulation can potentially be used at several levels of entity representation, its primary focus is at the platform level with the front line warfighter in the loop. As long as the exercise, test, or training involves warfighters in vehicles, DIS implementations will have to run at wall clock speed. The ultimate objective of DIS is to grow the capability to conduct entire theater level operations in real time at the vehicle level of representation and without great expense. The DIS standards are focused in that direction.

As depicted in Figure 6-1, DIS is currently operating at the platform level of aggregation which includes vehicles and other rigid bodies such as bridges, aircraft, and artillery pieces. Where units such as platoon or companies are depicted, they are displayed and simulated at the vehicle level. This includes elements such as convoys or the battalion Tactical Operations Center (TOC), which are made up of multiple vehicles.

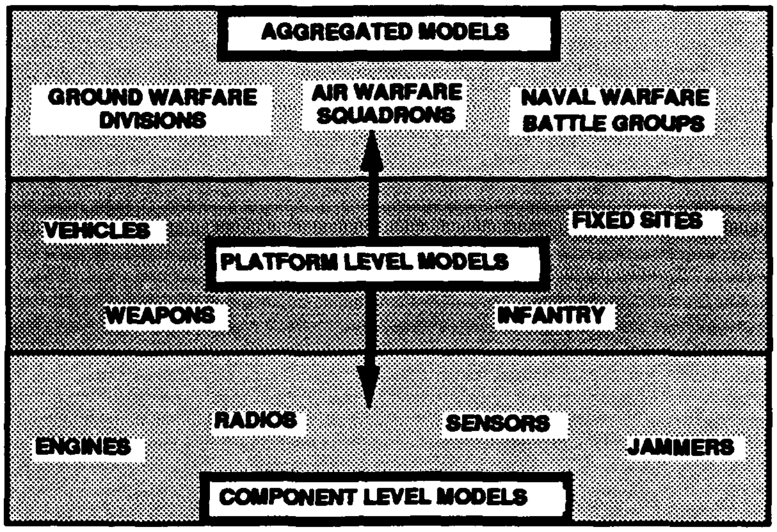


Figure 6-1. DIS Regime Operates Primarily at the Platform Level

The DIS architecture document recognizes that there are related standards processes ongoing in related simulation communities such as the Modular Simulation (MODSIM) project, the Joint Modeling and Simulation System (JMASS), and the Aggregate Level Simulation Protocol(ALSP). Each of these efforts is generally working horizontally to link elements within a given level of the available ways of simulating objects on the battlefield, namely aggregated models, platform level models, and component or part level models.

Modular Simulation (MODSIM) deals with simulation of components and modules which make up platforms such as propulsion systems, weapon systems, sensor systems, etc. MODSIM is a standard used to provide commonality for data on data busses internal to vehicles and other platforms. As such, it operates at a level below the current DIS protocols. However, there is no reason not to define a standard interface between the levels. DIS uses certain modules in conjunction with its vehicle level platform simulations, especially radios, sensors, jammers, etc, but it is not known how many DIS users will require component level entities. There is nothing in the DIS architecture that precludes operating at the component level of detail and it may be appropriate for engineering and testing applications within the physical limits of the available network bandwidth. This is especially true for components of platforms which have well defined and very rapid interfaces such as IFF interrogators and IFF responders or radars and jammers.

The Joint Modeling and Simulation System (JMASS) project spans the component and platform levels in the air defense world. The JMASS objective is to build both generic and specific software objects that can be individually placed in a library and then assembled into simulated weapon systems as needed. Since many of these software objects operate in the electromagnetic spectrum, it is a natural opportunity to interact. However, it must be kept in mind before the efforts are joined by decree that the objective of the JMASS effort is to build very high fidelity emulators which in many cases run slower than real time and do not have a man in the loop.

At the aggregated unit simulation level, organizations are represented without individual representations of their vehicles or personnel. Battalions and divisions fight equivalent units using attrition algorithms that have been developed to represent aggregated combat results. Time steps and terrain resolution are also aggregated. Mixing DIS platform level and aggregate unit level simulations can be done in three ways. The simulations can be allowed to interact at different levels of aggregation. However, this will require developing yet another set of algorithms in addition to platform level interaction or unit level interactions. Consequently the most likely approaches involve either aggregating the platforms into their respective units or deaggregating the simulated units into their component platforms. The latter requires templates for vehicles layouts, sets of models at the vehicle level, and detailed representations of terrain all displayed continuously in three dimensions. A far easier task involves aggregating a DIS platform level force into an appropriately sized aggregated unit and adopting the movement and attrition models in use within the higher level model.

The ALSP program is working to provide a set of DIS-like protocols for interfacing dissimilar models at the aggregate unit level which could support interaction if the DIS platform level model were aggregated. DIS is simultaneously investigating interfaces of its vehicle level simulations with higher level models such as the Army Corps Battle Simulation (CBS), Corps Battle Analyzer (CORBAN), and EAGLE with the intention of deaggregating the higher level models into individual platforms. The results of these efforts will determine how well and how soon DIS can interface with higher level models without major changes in either the higher level unit models or the DIS platform level models. Regardless of the outcome, it appears that the closer aligned the ALSP protocols are with the DIS protocols, the easier it will be to interface the two levels.

6.3 DIS Policy Issues Related to Data Bases

There are advantages and disadvantages to any implementation selected for DIS. However, at some point, a decision must be made on the basis of the best available information and a course of action selected. This is the case with the terrain data base standard being produced by Project 2851. DIS cannot wait for development of a perfect standard. So many things have to be done that policy should dictate pressing on. It is recognized that mistakes may be made, but it is vital that progress proceed as rapidly as possible to bring implementable and effective standards to the DIS process before more simulators are built.

It is envisioned that there will only be a few SIMWORLD databases in the DIS library. This is an arbitrary decision, but it is critical to husbanding the available DIS M&S resources and applying them to a relatively small set of common environments. Without enforcing this limitation on the number of unique SIMWORLD data bases through policy, we leave DIS open to rapid dissolution as each simulation community, service, and system developer attempts to create its own environment. This is a continuation of the current process and almost always results in incompatible simulations. Consequently, the adoption of the concept of the DIS SIMWORLD database standard along with policy which places reasonable limits on the number of SIMWORLDs to be built is central to the success of DIS at this stage.

There are many organizations with charters to collect data for models and simulations (and lots of other purposes). As DIS moves to standardize its data as well as its messages, some decisions have to be made concerning the sources and formats of the data and the means to collect, maintain, and distribute the data. To be successful, DIS must have a single data dictionary, a library that provides well configured data to authorized users, and a configuration management system to maintain the dictionary, the library and the public software associated with the standards. The library would contain certified copies of data such as:

- Project 2851 terrain data
- Weather data
- Electromagnetic models
- Weapon system models and parameters
- Damage tables and algorithms

At the present time, such an infrastructure does not exist and the responsibility for establishing and funding it has not been determined.

6.4 DIS Policy Issues Related to Software

The architecture envisions three types of software being maintained to support DIS. The most common type will be the highly configured and validated software which makes up the DIS-generated standard models, algorithms, etc. The second type is prototype software developed under various research or study programs and which is maintained in the library primarily for the purposes of reuse by other researchers working in related areas. The third type is proprietary code related to specific commercial standards that the DIS community either decides to adopt or is using as an interim product pending development or approval of a general standard.

A key question concerning software is reuse. To achieve software reuse, standards must be developed for software, and software libraries must be established. This architecture encourages, but does not ensure reuse of software.

The best way to promote reuse is to develop a policy to supply and promote the use of DIS software & documentation (preferably in electronic format) with Requests For Proposal (RFPs) and direct that certain functions and representations of the environment be used unless the contractor is willing to fund an independent verification and validation effort. If the library is created, large numbers of vendors may want to place software into the DIS library. Consequently, a policy will be needed on placing commercial, but DIS certified software products in the library.

While reusable software is a major aspect of the DoD policy on modeling and simulation, there must also be an acknowledgment that DIS is based on a rapidly advancing technology. Thus, it is too early to standardize on specific software modules as ways of producing algorithms, generating PDU messages, or implementing specific models. At this stage, both the standards and the technology are too dynamic.

A related software policy topic is the ability of the Ada language to support full object-oriented implementations. The Ada 9X committee is considering object-oriented extensions to Ada, but there is no indication at this time as to when such extensions might become part of Ada. In the meantime, there are commercial software products that provide these extensions. Policy must be developed that balance the general requirement for Ada with the current Ada limitations in producing object-oriented code.

6.5 DIS Policy Issues Related to Access and Payment for Use

While the question of which user has priority of use of DIS general resources is currently being addressed in Range Commanders Conferences, it is anticipated that policy will have to be issued to support coherent planning for the existing and future DIS facilities and the DIS network (should one be permanently established).

A related topic is the question of appropriate charges for the users of the general purpose DIS resources. At the present time, the Laboratory Support Environment task under the Advanced Distributed Simulation Technology contract maintains the two Army BDS-D sites and provides a minimum level of support to specific projects using the BDS-D facilities. It is envisioned that the expansion of DIS will eventually be self supporting with those organizations using common DIS facilities paying a share proportional to their use. It is anticipated that this procedure will also be applied to DIS network charges if a pay-as-you-go network is set up.

6.6 DIS Policy Issues Related to Openness

6.6.1 DIS Meetings, Conferences, and Communications

There is an unwritten rule that access to DIS information will be free and open. Consequently, the DIS meetings have been unclassified and open to everyone with the attendance fee. On the other hand, as time progresses, DIS will increasingly represent an accumulation of technology which has national strategic value for training and readiness. Procedures have already been put in place to review the requests of each potential user of the DIS Bulletin Board System to minimize the chance that unauthorized personnel obtain access.

6.6.2 DIS Contractor Support

There are several contractors involved in developing DIS. Each is supposed to make available all DIS work conducted at Government expense. This should be codified in writing rather than left to interpretation.

6.6.3 DIS User Community

Another unwritten rule is that government and industrial organizations conducting unclassified experiments which use some part of the government supported DIS laboratory resources should share the results of that work with the DIS community. If that is not their intention because the work is sensitive, solicitation related or proprietary, then arrangements should be made in advance to restrict access. This policy should be codified in writing rather than left to interpretation.

6.7 DIS Policy Issues Related to Funding the Infrastructure

Without funding for infrastructure, standards are just pieces of paper. The bottom line is that a considerable investment is required to create and maintain the DIS infrastructure. That infrastructure includes

- The standards and architecture process
- DIS Library (SIMWORLD and Battlefield data bases and supporting documentation)
- Conformance System & Benchmarks
- Technical Infrastructure & Laboratory Support Environment
- Baseline Network

The payoffs resulting from standardization of the several virtual environments and the provision of networking will be evident almost immediately. DIS will reduce costs for new training devices due to more rational standards and reusable code and components. It will also significantly enhance readiness for the training community and analysis and test capability for the acquisition community.

7 Security Considerations

7.1 Overview

This section discusses important reasons for DIS to be concerned with security engineering issues, including the need for performing classified processing with unclassified output. DIS is international in scope, and the security ramifications of operating with multilevel and multinational data will require special care in terms of access control and accountability. For this reason, establishment of a DIS architecture that supports the needs of security accreditation, granted by a to-be-identified Designated Approval Authority (DAA), will result in a smoother transition into classified DIS operations.

This section is organized into three main parts. Subsection 7.2 deals with the types of security rules that need to be enforced by DIS. Subsection 7.3 identifies and describes some components that can be configured in a variety of ways to achieve a security architecture satisfying the security rules. Subsection 7.4 shows several examples of such architectures, based on a variety of likely DIS scenarios.

7.2 Important Computer Security Policy Concepts

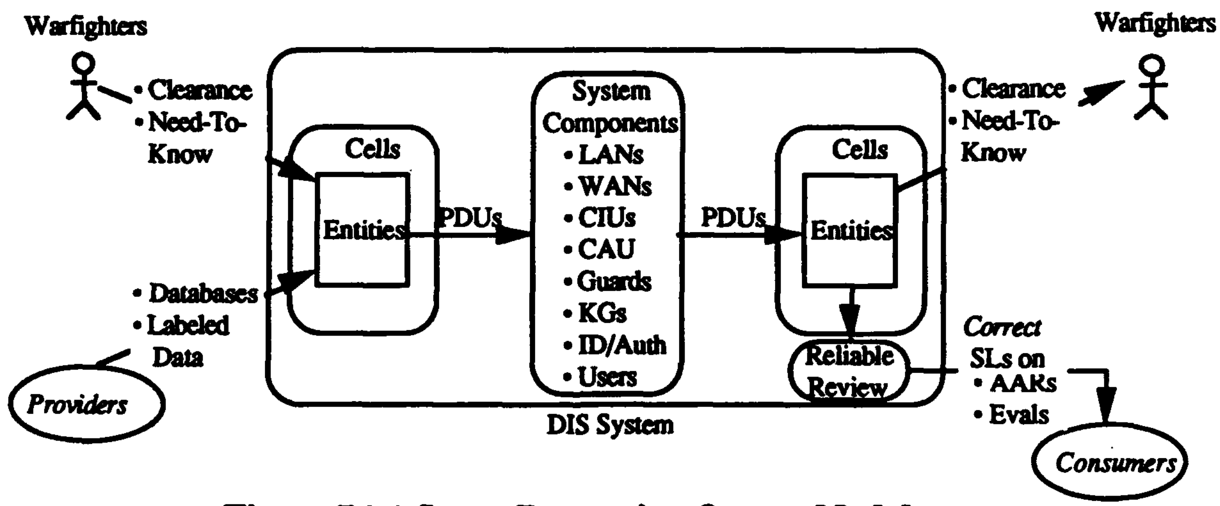


Figure 7.2-1: Input-Processing-Output Model

Figure 7.2-1 presents a top level input-processing-output view of DIS, which is a useful starting point in stating the security considerations that need to be taken into account. One or more of the standard DIS cells processes, stores and protects classified information, so security design considerations are important. *Warfighters* represent inputs which arrive with known security credentials—they are either uncleared, or their clearance and briefing status is specified. The nature of their warfighter role will define their need-to-know characteristics. These attributes remain unchanged as they become system outputs. *Providers* (those who provide system-level DIS inputs) supply databases with which DIS interacts. These inputs must be associated with correct *explicit* (provided with the data) or *implicit* (derivable via a process) Sensitivity Labels (SLs). Note that *correct* SLs are the same as those which would have been applied by the classification authority for the information, as compared with *safe* SLs—those which dominate the information's true security sensitivity. Report products, destined for *consumers*, (those who receive system-level DIS outputs) are required to be correctly labeled by design of the DIS system.

The ultimate security properties of DIS require a clear, concise, DIS-specific statement of rules, which will be presented in the DIS Computer Security (COMPUSEC) Policy (hereinafter called the COMPUSEC Policy). The COMPUSEC Policy will consist of three main sections:

- Technical Definitions (concise meanings of important terms used in the COMPUSEC Policy)
- Assumptions (rules whose enforcement is beyond the control of the project)
- Policy Statements (rules whose enforcement is within the purview of the project)

The hardware, software and firmware mechanisms that an implementer designs and builds to enforce the COMPUSEC Policy's rules is known as the Trusted Computing Base (TCB).

A number of technical terms require a clear definition. These terms have specific meaning within the security engineering community and need to be used consistently within DIS. Some of these terms include: correct, safe, validated, sensitivity labels, lattice, sensitivity of information, information security perimeter, export.

Sample Assumptions are as follows:

- The three types of databases that enter a standard cell (i.e., SIMWORLD, Battlefield, Session) are appropriately validated prior to being loaded into the standard cell.
- People entering a standard cell are appropriately cleared and briefed, based on the duties they will be performing and the security level at which the cell is accredited.
- SLs on database records entering a standard cell are correct.

Sample Policy rules are as follows:

- There is a population of SLs, known as the DIS Security Lattice, that is used to label subjects and objects within DIS.

- Within the DIS System, it is only permissible for an object to be labeled with an SL whose value is greater than or equal to the sensitivity of the information contained within the object.

- Reports generated within DIS and exported across the DIS Information Security Perimeter (defined below) are labeled correctly.

- The ability for a subject to access an object within DIS is based on the subject passing appropriate Mandatory Access Control and Discretionary Access Control rules. Such rules, in turn, are a reflection within DIS of the types of rules that the subject would need to pass in the "real world" to access the actual object.

7.2.1 Information Security Perimeter (ISP)

"Perimeters" delimit system properties. The DIS system perimeter, for example, is the outermost boundary in Figure 7.2-1, and is needed to distinguish inputs versus outputs to DIS. The ISP is an important security engineering boundary. It delimits a secure system's TCB since the COMPUSEC Policy is enforced within. The ISP provides design and discussion focus when dealing with DIS security issues.

7.2.2 System Inputs

System inputs to DIS include the following:

- Warfighters, i.e., the people who operate the simulation equipment, and who need access at least past the physical security perimeter. Warfighters may also need access to the information protected within the ISP, and so their presence and actions are security-significant.

- Databases: Typically, a database is composed of tables, which, in turn, are composed of records, which, in turn, are composed of fields. Depending on DIS needs, SLs can be associated at any of these levels of granularity. As a strawman position, DIS objects are assumed to be labeled at the record level. The purpose of these labels is to allow the TCB to enforce MAC rules based on the label of a subject attempting to access a given database record. In addition, DIS might need SLs associated with higher level entities, such as a database table. Generally a table SL would not be used directly in enforcing rules concerned with a subject accessing an object, but would instead be used to assure that records stored in the table contain SLs that are dominated by the table SL. There is need for clear COMPUSEC Policy guidance regarding TCB enforcement mechanisms (e.g.,nonviolable binding of a record with its SL, assurance that the SL value always dominates the information within the record.

An example is a database containing information on a new weapon using a long-range projectile. Although technical details of the weapon, and possibly even the existence of the weapon itself, might require classification (i.e., association of SLs with records within the weapon's database), it is possible that unclassified PDUs could be generated which simply indicate whether or not some target has been successfully hit by the weapon.

7.2.3 System Outputs

Each of the DIS user communities (e.g., training and operational exercise support, test and evaluation, combat development, materiel development) have their own specific goals in using DIS. This section discusses DIS outputs, at the level of abstraction represented in Figure 7.2-1, that might be generated based on any of these user communities engaging in a DIS exercise, from a security perspective.

Typical DIS system-level outputs include Data Logger tapes, After Action Review (AAR) reports, warfighter evaluations, and evaluations of DIS itself. Each of these outputs can exist in hardcopy or softcopy. In particular, the COMPUSEC Policy's rules need to correlate the SL of the output with the SLs of the various inputs which were accessed in creating the given output, as well as the trustedness of the algorithm which generated the output based on the values of the inputs. As stated earlier, an important COMPUSEC Policy rule will require correct SLs on each output.

7.2.4 Mandatory Access Control (MAC)

MAC deals with the ability of a subject to access an object, based on the values of the SLs associated with each. Typically, a subject is permitted to read an object only if the SL of the subject dominates the SL of the object. For modifying an object, this is normally reversed; real-world projects, however, have found this latter rule too simplistic, and generally state the MAC rule for a subject to modify an object along the following lines: A subject may modify an object only if the clearance of the subject dominates the SL of the object, and the current SL of the subject is dominated by the SL of the object. Formulating MAC rules in this fashion makes it possible to demonstrate the important security consideration that information never flows "downward" in sensitivity, e.g., that it is not possible for SECRET data to be received by CONFIDENTIAL subjects. Mechanisms to guard against such an occurrence need to become part of the DIS TCB.

7.2.5 Sensitivity Label (SL)

The preceding discussion has demonstrated the need for SLs to be associated with subjects and objects in terms of stating and enforcing MAC rules. It is the responsibility of providers of inputs to DIS to assure that SLs are present on all information requiring SLs. Furthermore, since DIS can not be the classification authority for data presented from outside the system, providers are responsible for insuring that the value of the SLs on their inputs are correct.

Within an Information Security Perimeter, SLs are associated with PDUs, in accordance with security rules that will need to be stated initially in the COMPUSEC Policy, and refined in the COMPUSEC Policy Model. The Trusted Computing Base enforces PDU flow such that PDUs never flow downward. Within a standard DIS cell, the TCB enforces the rule that all SL values are dominated by the permissible maximum for that cell.

7.2.6 Reliable Review (R^2)

For information intended to be exported from within the ISP, it is necessary to assure that the SL associated with the information is correct, rather than being merely safe. R^2 is a method to assure this property. R^2 may be viewed as a transform with inputs and outputs, subject to the fact that certain preconditions need to be true for the outputs generated.

The inputs to R^2 include the information intended to be exported across the ISP, the SL associated with the object containing the information as generated automatically by the TCB, and the nominated value of the SL which is felt to be the correct value. The preconditions of R^2 would verify that the nominated value is dominated by the current value of the SL. The processing part of R^2 verifies (either automatically, or with a person in-the-loop) that the nominated value of the SL does correctly represent the sensitivity of the information. If so, the output of R^2 is the information with its SL set to the nominated (correct) value, and the information is routed for exportation across the ISP. If the nominated value is deemed not to be a correct value for the information, then R^2 is failed for this information, and error processing needs to be performed.

7.2.7 Discretionary Access Control (DAC)

A subject needs to pass both MAC and DAC checks to be permitted access to an object. DAC checks are based on security parameters other than SLs; DAC information needs to be associated with both subjects and objects, and there need to be TCB rules stating the logic which is used to grant or deny DAC access based on the values of DAC information associated with the given subject and object.

For a subject, the needed DAC information is typically either the individual identity of the subject, or the user groups to which the subject has been assigned. For example, a Combat Service Support (CSS) warfighter may be cleared to the SECRET level at which the cell containing the Tactical Operations Center is operating, and thus would pass MAC checks for access to local databases contained within the entity. However, his role (as a CSS operator) does not demonstrate a need-to-know for such data. Therefore, access control mechanisms, part of the DIS TCB and under the control of the site's Information System Security Officer, would deny any attempted access.

DAC also is used for multicast services. In contrast with broadcast, in which PDUs are sent to all entities participating in an exercise, a multicast PDU is sent only to entities belonging to the same multicast group as the PDU originator, analogous to user groups in traditional DAC.

7.3 Architectural Components

The DIS security architecture relies upon components which work together to enforce the COMPUSEC Policy. These components comprise the DIS TCB. This section describes the types of COMPUSEC Policy enforcement performed within a variety of DIS architectural components.

7.3.1 Identification and Authentication

Identification, and authentication of the claimed identity, are important parts of the TCB since several COMPUSEC Policy rules are tied to special knowledge about a subject's identity as it attempts access to objects. To ensure DIS participants include only known simulation cells, an authentication component will identify and authenticate cells which attempt to utilize resources and establish or participate in an exercise. The authenticator, which need not be centralized, will also maintain exercise-specific participation records as part of the system's audit trail.

7.3.2 Policy Enforcement Roles of People

Security compromise is likely if people associated with DIS behave maliciously, or commit errors of omission or commission while interacting with DIS resources. Individuals specifically counted upon for enforcing one or more of the COMPUSEC Policy's rules (e.g., Information System Security Officers, who maintain access control databases; exercise controllers, who configure virtual network connections between standard DIS cells) are termed "DIS users," and they require specific training in the correct performance of their security-related duties. In contrast, "DIS participants," including those participating in classified exercises, have access to simulation equipment controls and displays, but are unable to alter secure operation of access control mechanisms in the system's TCB.

TCB design will be focused to generate and protect exercise-specific audit trails of DIS user actions to produce a picture of events leading up to security anomalous conditions.

7.3.3 Virtual Network: Carries Unclassified Traffic

A basic DIS objective calls for use of commercial facilities to implement the system's Virtual Network-Intercell Tier (VN-IT). Therefore, traffic on the VN-IT must be unclassified. The DIS security architecture embraces this requirement using COMPUSEC Guard mechanisms and encryption devices, as appropriate, between the VN-IT and cells processing classified data. Note, however, that this same prohibition does not attach to the Virtual Network-Cell Tier (VN-CT). Where appropriate to the simulation, based upon site-specific needs and conditions, the VN-CT may transport classified information within the cell between its various entities.

7.3.4 Computer Security Guard Mechanism

Standard DIS cells which process classified information but produce PDUs intended for broadcast or multicast via the VN-IT require these PDUs to be certified unclassified. The COMPUSEC Guard mechanism acts as an automated "classification authority" for unclassified PDUs. The level of assurance required by the COMPUSEC Guard mechanism is a function of the risk of compromise it must control, which itself is a function of the security level at which the cell it is guarding is operating.

7.3.5 Cryptographic Devices

Standard DIS cells which process classified data need a mechanism through which they can exchange classified messages over the VN-IT. Crypto devices, such as KGs, allow classified cleartext (red data) to be handled as unclassified cyphertext (black data) for transmission. KGs will be located within CIUs, for standard DIS cells, or CAUs, for nonstandard DIS cells, connected down-stream of other required CIU or CAU processing. SLs are not part of the header of messages intended for transmission over the VN-IT, but may be contained within the body. The CAU or CIU will strip header data from an outgoing message, including routing indicator, and encrypt the balance. The receiving CAU or CIU will reconstitute the message prior to delivery to the cell. Cryptologic key management is a security-relevant issue. Established key distribution techniques will be implemented. Automated key distribution schemes, perhaps utilizing STU-III telephones, may become available.

7.3.6 Standard DIS Cells

Some percentage of standard DIS cells deal with simulations using unclassified data. Others are associated with development programs which involve classified data, with databases often containing special performance characteristics. The DIS security architecture will be responsible for preventing the VN-IT from becoming contaminated with classified information which originates within a cell processing classified data. The COMPUSEC Guard mechanisms and cryptologic devices discussed above will provide necessary protection. However, like all other communications devices, there will be some contribution to latency of the PDU being processed.

7.3.7 Cell Interface Unit and Cell Adaptor Unit

CIUs and CAUs connect either standard or nonstandard DIS cells to the VN-IT. Where cells are exchanging classified data, the CIUs and CAUs contain important elements of the DIS TCB. These include the COMPUSEC Guard mechanisms and cryptologic devices described above. CIUs and CAUs can also participate in DAC by filtering arriving PDUs which are not addressed to the cell.

7.3.8 Nonstandard Cells

Nonstandard DIS cells are simulators or other environments which do not adhere to DIS connectivity standards. From a security engineering standpoint, they can freely participate in DIS to the extent their behavior fits within the environmental bounds described by the COMPUSEC Policy's assumptions. For example, a nonstandard cell, which processes classified data, can be connected via a CAU to the VN-IT, provided it correctly labels its messages. Mechanisms within the CAU could then convert unclassified non-DIS messages into cleartext PDUs, or produce black cyphertext if the non-DIS message was classified.

7.4 Combining Components into a Security Architecture that Enforces the COMPUSEC Policy

7.4.1 Classified System

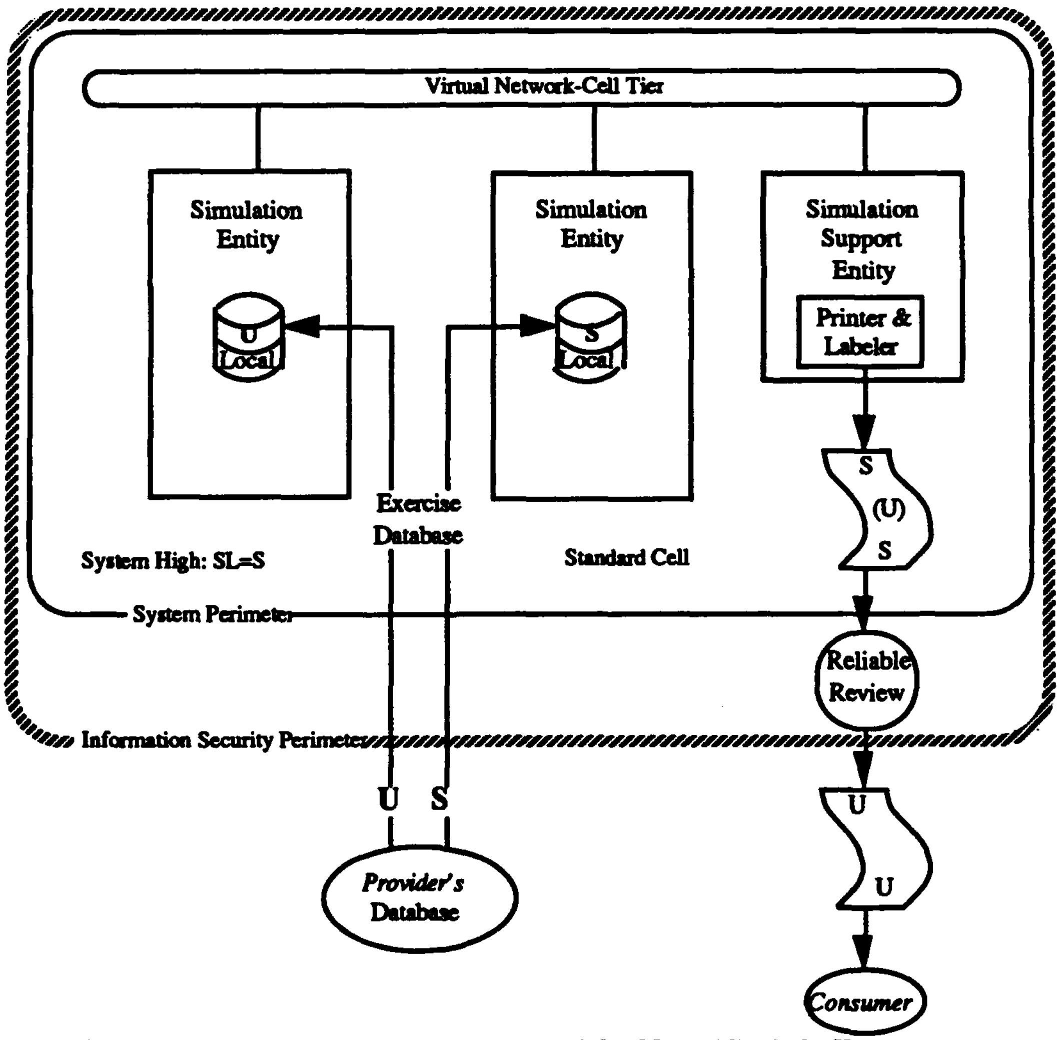


Figure 7.4-1: DIS Architecture With Classified Cell

Figure 7.4-1 illustrates a single-level standard cell working with SECRET level data. The following COMPUSEC Policy concepts are illustrated:

- The ISP contains a set of homogeneous components, all working with data that is protected at the cell's High-Water Mark (HWM) SL.
- The input databases are correctly labeled with an explicit machine-readable or human readable SL. Containers, including communication lines, all protect data commensurate with the HWM SL. Copies of the database retain the SLs of the original database. One way to maintain the SLs is to place unclassified data into one storage system and classified data into a separate storage system. These storage systems are examples of containers with implicit SLs which can serve as upper bounds on the levels of data they contain.
- Output reports are protected and labeled at the HWM SL of the cell (SECRET in this example) regardless of the level of the contents (which may, for example, be unclassified). Unclassified data, labeled SECRET, is an instance of safe labeling. R^2, performed by a manual process outside the System Perimeter, but inside the ISP, changes the safe SECRET label to a correct unclassified label. Though not specifically illustrated, the same process applies to the production of classified reports (e.g., correct CONFIDENTIAL label replaces safe SECRET label following successful R^2).
- Warfighters and others with unescorted access to the cell are cleared and briefed to the level of the data protected, processed and stored within the cell. They must be identified and authenticated by a mechanism before being granted access to data within the cell. DIS users must have "need-to-know" for some of the information contained within the cell, and DAC mechanisms control their access.
- Entites send DIS PDUs to each other via a cell broadcast mechanism. DAC mechanisms within an entity can limit distribution to DIS users and other subjects with the proper need-to-know.

7.4.2 Distributed System with Classified Cells

Figure 7.4-2 illustrates two distributed classified cells. The following COMPUSEC Policy concepts are illustrated:

- All data within the cells is treated as classified at the HWM SL at which the cell is operating. The VN-IT contains only unclassified information.
- CIUs contain an encryptor which is used to facilitate non-PDU communications between cells via the VN-IT using point-to-point encryption. When several cells are connected, several point-to-point encryptors can be used, or an SDNS product can be used. SDNS products use encryption between layers 3 and 4 in the ISO model. All

data above those layers are encrypted. This means that the DIS PDU is encrypted. All header information and address information below those layers appear as cleartext on the VN-IT. Only cell and exercise addresses are in cleartext. Within a cell, at a given site, PDUs are broadcast over the VN-CT.

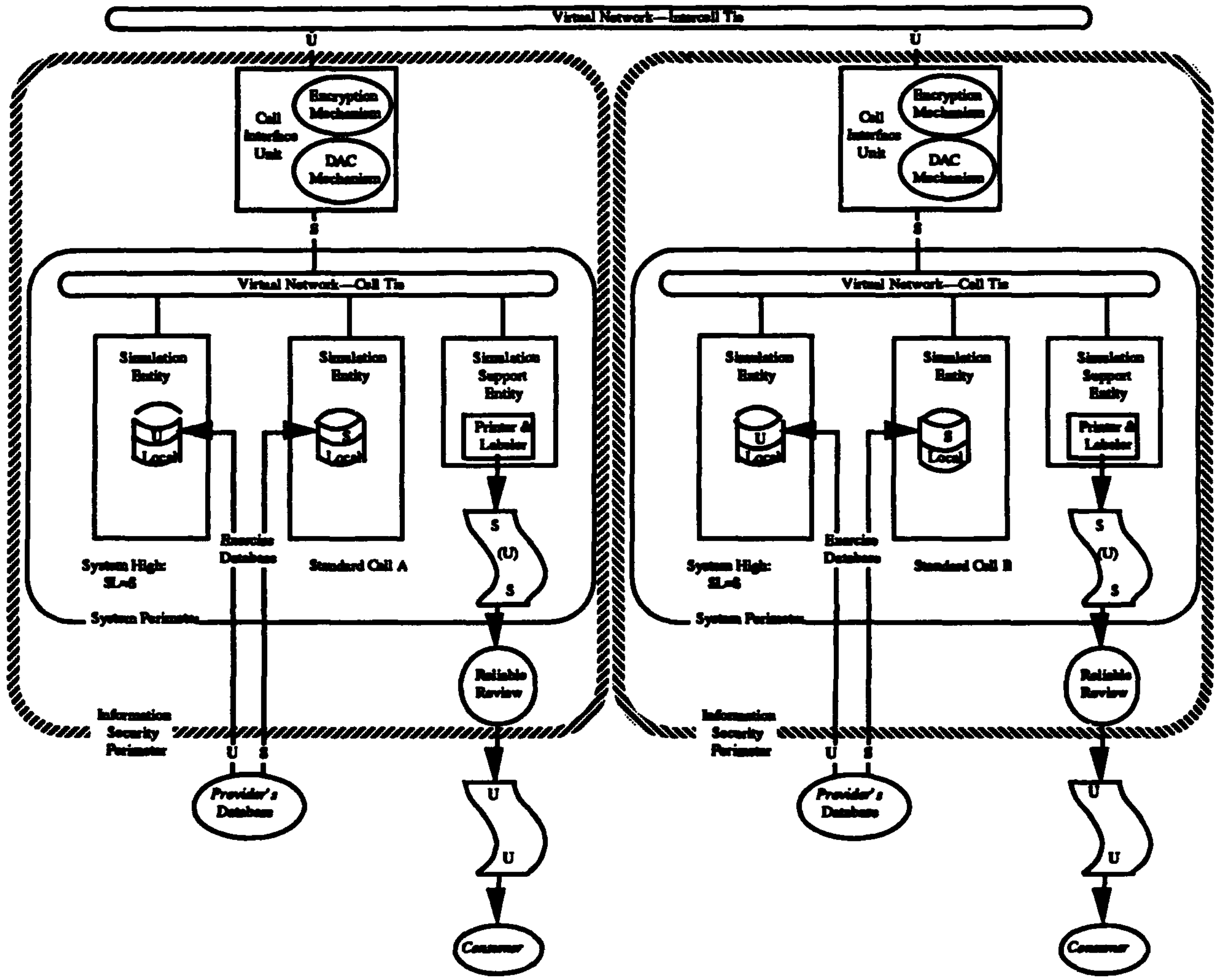


Figure 7.4-2: DIS Architecture with Distributed Classified Cells

- As in the previous example, output reports are protected and labeled at the HWM SL of the cell (SECRET) regardless of the level of the contents (which may, for example, be unclassified). Unclassified data, labeled SECRET, is an instance of safe labeling. R^2 is performed by a manual process outside the System Perimeter, but inside the ISP, R^2 changes the safe SECRET label to a correct unclassified label. Once more, though not specifically illustrated, the same process applies to the production of classified reports (e.g., correct CONFIDENTIAL label replaces safe SECRET label following successful R^2).

- Warfighters and others with unescorted access to the cell are cleared and briefed to the level of the data protected, processed and stored within the cell.
- Individuals that change the encryptor keys must have proper COMSEC clearances.
- A DAC mechanism in the CIU can be used to limit PDUs sent to the remote cell.

7.4.3 Distributed System with Mixed Cells

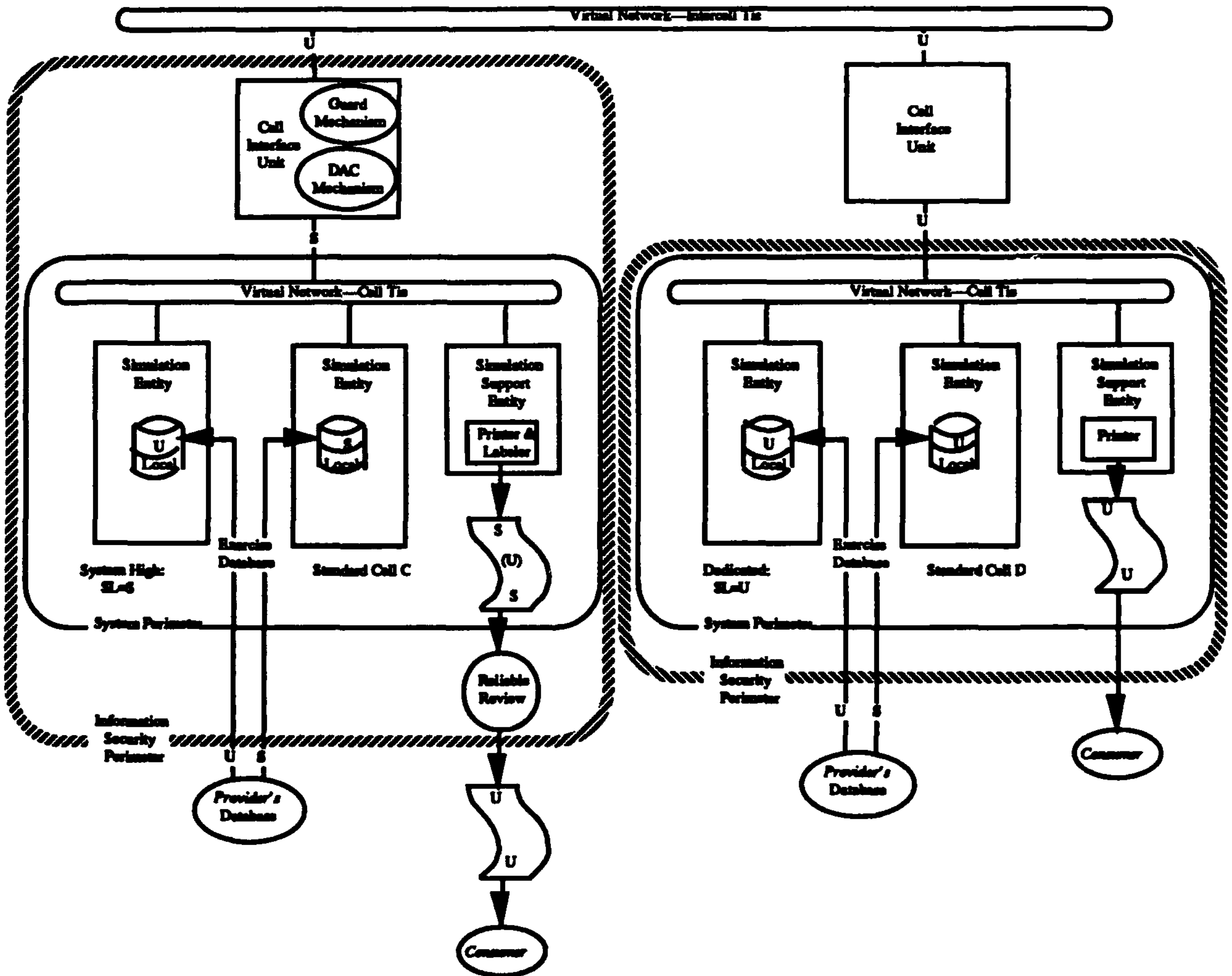


Figure 7.4-3: DIS Architecture with Distributed Multilevel Cells

Figure 7.4-3 illustrates two cells, one processing unclassified data and the other processing SECRET data, in the same exercise. This might be an exercise to evaluate a new weapon with a new long range projectile. The range and ballistics of the projectile are classified. PDUs from the classified cell, which it asserts to be unclassified, are checked by the COMPUSEC Guard mechanism in its CIU to ensure only unclassified data is delivered to the VN-IT. An unclassified PDU could tell a remote entity that it is hit without providing any classified information (unless the hit notification conveyed a classified fact related, for instance, to weapon range, which would render the fact unable to be transmitted as unclassified). The following COMPUSEC Policy concepts are illustrated:

- The classified cell and the unclassified cell exist within separate ISPs, since the rules enforced within each cell are different.
- The classified cell's CIU contains an automated COMPUSEC Guard, which functions as described above. There is no encryption mechanism in this example since only unclassified PDUs are passed between cells.
- Output reports from the unclassified cell are correctly labeled unclassified, and therefore do not need a R^2 mechanism However, output reports from the classified cell must protected and labeled at the HWM SL of the cell (SECRET) regardless of the level of the contents (which may, for example, be unclassified). As in previous examples, R^2, performed by a manual process outside the System Perimeter, but inside the ISP, changes the safe SECRET label to a correct unclassified label. As before, though not specifically illustrated, the same process applies to the production of classified reports (e.g., correct CONFIDENTIAL label replaces safe SECRET label following successful R^2).
- As in previous examples, warfighters and others with unescorted access to the cell are cleared and briefed to the level of the data protected, processed and stored within the cell (which in this example means they need no clearance at all for access to the unclassified cell).
- Once more, individuals that change the encryptor keys must have proper COMSEC clearances.

7.5 Conclusion

It has been shown that the DIS security architecture revolves around components which work in concert to enforce rules to be presented in a COMPUSEC Policy. The COMPUSEC Policy will embrace the need to use commercially available networks to implement the VN-IT as a carrier of traffic which must be maintained at the unclassified level. COMPUSEC Guard mechanisms and cryptologic components provide assurance that classified data will not be compromised during VN-IT transmission. MAC mechanisms will operate to protect against write-down and read-up COMPUSEC Policy violations, and DAC mechanisms will enforce need-to-know and need-to-do concepts. The security architecture allows multiple levels of classified messages to be exchanged during simultaneous exercises, and is compatible with the security needs of DIS.

8 Verification, Validation, And Accreditation

8.1 Introduction.

Model requirements imposed by the model verification, validation, and accreditation (VV&A) needs of the BDS-D will impact design and implementation of both the public and private sections of each participating model/simulator, as well as the protocols of the distributed network. The BDS-D must be implemented in such a way as to provide readily accessible testpoints within each model/simulator. This allows the performance aspects of the simulator relevant to a specific experiment to be measured. There must be means for communicating these measurements to the validation/accreditation authority. While off-line communication via magnetic media is feasible, communication using the BDS-D distributed network may be far more useful. For a large-scale, multi-purpose, battlefield simulator, model VV&A will be an activity conducted during its entire useful life, hence provision for conduct of these activities efficiently and effectively must be a part of the BDS-D architecture and design.

8.1 Definition of Model VV&A

The BDS-D is generally presumed to be comprised of computer software-based models/simulators interacting amongst one another via a digital communications net. Even so, model VV&A is applicable to models implemented in ANY technology, and it must thus be distinguished from software verification and validation. The latter activities are applicable to any software system, model or not. If software V&V is performed, it is generally performed by an agency independent from the software development agency, it is intended to enhance the prospect that delivered software performs in accordance with software requirements, and software V&V activities generally terminate upon acceptance of the software product. Software V&V only implicitly assesses the software product's suitability, concentrating instead on whether the software developer "performed as required by contract". Software V&V is a contributor to the model VV&A process, but it is not strictly necessary, nor is it sufficient.

By contrast, model VV&A is an activity performed during the entire life cycle of a model. A new VV&A effort is associated with each experiment that is performed on the model. Model VV&A is inextricably tied to the specific user and use of the model.

8.1.1 Verification

The process of determining that a model accurately represents the developer's conceptual description and specifications. In a large scale model development, verification is applied at each stage to ensure that the products of the stage accurately implement the specifications from the previous stage.

8.1.2 Validation

The process of determining that a model is an accurate representation of the intended real-world entity from the perspective of the intended use of the model.

8.1.3 Accreditation

The process which certifies that a model is acceptable for specific types of application.

8.2 The Model Validation Process.

Architecturally, for a software-based model, software V&V activities contribute to model verification activities, and model validation and accreditation activities are completed only in the context of a specific user with a specific purpose. Economically, this may be an unacceptable burden on the model user or developer, and the architectural operational requirements dictated by model VV&A are largely dictated by the goal of performing as much as possible of the model VV&A process once, rather than once for each model use. Accordingly, we introduce the concept of model characterization as an intermediate step in the model VV&A process:

That part of the model validation process that measures the performance of the model at each point for which there exists an equivalent performance measure from the real-world counterpart. Model characterization specifically excludes assessments of the adequacy of the model, because characterization can precede definition of the requirements for a particular experiment.

In general, model characterization will be incomplete, in that some model and/or real-world performance data will be missing for any specific application. But model characterization is durable, in the sense that it remains accurate at least until the model or the counterpart real-world system is revised.

Model validation, then can be decomposed into the two steps of model characterization and comparison. The comparison of the model characterization data with real-world counterpart data will generally yield a difference (after all, it is a model). It is the magnitude or distinctiveness of the difference that, when compared with a user's explicit requirement, determines whether a model validly represents the system for the specific experiment. In general, a model will be valid for a number of factors, invalid by some margin for some other number of factors, and indeterminant (for instance due to a lack of real-world data) for some still other set of factors.

8.3 Technical Challenges.

Technical challenges associated with BDS-D model verification, validation, and accreditation (VV&A) requirements fall into two categories:

(1) Challenges related to the distributed simulation technology.

(2) Challenges related to data quality for model characterization.

8.3.1 Distributed Simulation Technology.

It is desirable to conduct VV&A of the BDS-D models and simulators within the framework of the distributed architecture.

In particular, it is desirable to infer internal functioning of important aspects of any participating model from the publicly visible manifestations of its participation -- that is, the data it generates and places on the BDS-D network. Currently defined protocol data units (PDUs), as documented in the draft DIS standard, do not directly accommodate measurement or even existence of many model behaviors critical to most validation and accreditation issues for BDS-D. The most desirable mechanism to compensate for these deficiencies is to identify mathematical or statistical transformations from required data points internal to the model communications sequences that a valid model can be expected to generate, and then to identify the stimuli that will cause the internal data points to behave as anticipated when the model is exercised.

To the extent that this challenge can be successfully addressed, it will become feasible to consider automated and standardized means to validate or characterize BDS-D models within the architectural framework of BDS-D. The alternatives for VV&A are to invade the private (from the BDS-D standpoint) parts of a participating model in order to observe model performance data, to expand the size and scope of the model's public interface, or to rely on accurate reporting of characterization data by the model proponent.

Another challenge related to the distributed architecture of the BDS-D concerns the dynamic configuration changes in the actual participants from one experiment or exercise to another, and potentially even, within a single exercise. Especially under these circumstances, the question of whether a model inherits validity from its submodels is of interest. In general, the answer is that model validity is not inherited from its submodels. This does not preclude identification of circumstances that permit the inference of model validity from the validity of its submodels. If successful, the payoff is significantly faster, easier, and less costly validation.

The BDS-D will enjoy an object-oriented architecture. Many of the BDS-D models will employ object-oriented design techniques, and it can be hoped that they will be programmed in an object-oriented programming language. Object-orientation embraces inheritance as a relation among objects, and there may be circumstances under which the inheritance among objects can be extended to inheritance of model validity among families of interacting models.

8.3.2 The Problem of Messy Data

The validation of any model or simulator, distributed or not, requires identification of a data source suitable as a standard of comparison. Although under limited circumstances, that standard could be a previously validated model addressing the same subject matter, this is less than desirable. Certainly more attractive on the face of it is data collected on the performance of an actual system, preferably in an appropriate operating environment, or alternatively in a laboratory setting. For a battlefield simulation, laboratory data (i.e. field exercise data) is virtually the only available standard of comparison data. But for a battlefield environment, this data is very frequently messy. Instrumentation for much battlefield exercising is relatively imprecise. Multiple instrumentation frequently yields uncorrelated and even uncorrelatable results. Yet there is often no alternative source for standard of comparison data.

Emerging statistical techniques have shown some promise in allowing messy data to be correctly treated as representative of an underlying real- world phenomenon. These techniques can be used even in situations where no assumptions can be made about statistical independence, and where the form of the underlying distributions may not be gaussian. They hold promise in validation of BDS-D models, and can be expected to yield better quantification of the degree to which a BDS-D model represents its real-world counterpart in a battlefield environment. This in turn gives the model user better evidence upon which to base an accreditation decision.

8.4 The User's Role.

Model validation is completed for a particular set of user requirements by identifying specific requirements on the model, including threshold or accuracy parameters by which to compare model performance against real-world performance. For any given specific requirement, then, the model can be validated by noting that the measured model performance is within an accuracy delta, or above a threshold, with respect to the counterpart real-world measurement. Where model characterization data is missing, the user may have to perform a characterization of the specific parameter. This new performance data then becomes available to the next user that requires a model validation with respect to that parameter.

Any specific user of the BDS-D is likely to require validation of a specific simulator function that is not covered by the Blueprint of the Battle subfunctions. Such requirements shall be satisfied by the user as a private VV&A requirement. However, characterization data, both simulator and real-world, will be provided to BDS-D and will be archived for future use by other users.

8.5 The Role of PMTRADE.

The architecture for BDS-D must provide for convenient characterization of the model during its entire life cycle. A major architectural question is how general this characterization should be. A reasonable answer can be found by examination of the Blueprint of the Battlefield (TRADOC Pamphlet 11-9). The BDS-D is primarily a simulator of the tactical battlefield, with some aspects of simulation of the operational battlefield. It is possible that the Distributed Interactive Simulations (DIS) draft military standard could be expanded to include PDUs suitable to allow any model/simulation to report its implementation of each subfunction of the tactical battlefield blueprint as listed in Appendix D of TRADOC PAM 11-9, and selected subfunctions of the operational battlefield blueprint as listed in Appendix C. The architecture, the PDUs, and the model implementation could facilitate each participating model reporting whether the model implements a subfunction, and if it does, describing the model characterization vector for that subfunction.

The BDS-D developer should locate and provide real-world characterization vectors for each subfunction of Appendix D and the subfunctions of Appendix C determined to be relevant to the BDS-D simulation. Cases where data characterizing a real-world subfunction is not available are identified for subsequent test and measurement by an appropriate agency, such as TRADOC.

A Glossary

Actual Battlefield: The combat environment that simulation technology attempts to replicate. Successful simulation will cause the participating warfighters to act as if they were engaged in actual battle. The term real battlefield is often used synonymously with actual battlefield. Note that battlefield is used in the general sense, including air, land, and sea combat.

Aggregated: A term generally applied to unit models in which all platforms and vehicles cannot be individually distinguished. In addition to organizational aggregation, models can aggregate time (large time steps), space (gross resolution in sectors, hexes, boxes, etc.), and functions (unit level attrition, maintenance, etc.). The Aggregated Level Simulation Protocol (ALSP) is being developed to link dissimilar aggregated models. Note that platform models can be thought of as aggregations of modules and modules as aggregations of parts, etc.

Autonomous: A battlefield entity which does not require the presence of another battlefield entity in order to conduct its own simulation in the battlefield environment is said to be autonomous. All DIS compliant battlefield entities are autonomous.

Battlefield Data Base (BATTLEFIELD): Database which defines the specific domain of an engagement. It includes the parametric data needed to generate a version of the SIMWORLD which when combined with the SESSION data base can generate an exercise. The BATTLEFIELD in all caps is used in this volume as a shortened notation for "Battlefield Data Base".

Battlefield Entity: A simulation entity which corresponds to actual equipment, supplies, and personnel that can be seen or sensed on a real battlefield. Platform level battlefield entities include aircraft, ships, armor vehicles, dismounted infantry soldier, guided missile, command post, truck, etc. Unit level entities, such as platoons, companies, etc. can also be defined. A battlefield entity incorporates a direct soldier/machine interface which replicates the soldier/machine interface of the actual battlefield entity.

Cell: A cell is a set of simulation entities using fully consistent database and simulations, i.e. the simulation models have been specifically designed to work together. All entities within a cell must have unrestricted broadcast of datagram messages to all other entities within the cell. By definition the entities in a cell are homogeneous, and at the same security classification level. For example, a set of interconnected SIMNET simulators using the same terrain database constitute a cell.

Cell Interface Unit (CIU): A processing module which interfaces a DIS Standard Cell with the virtual network. One device is required for each standard cell. CIUs provide intercell services such as message filtering, translation of messages, data compression, aggregation and deaggregation of simulation entities operating at different representation levels.

Cell Adapter Unit (CAU): A CAU interfaces a non-standard cell with the virtual network. It is functionally equivalent to a CIU, except that it adapts non-DIS cells to the DIS network.

Cell Data Base: The union of the SIMWORLD, BATTLEFIELD, and SESSION data bases within a cell. The information a cell needs to configure itself for an exercise.

Common Data Base: A general term used to describe the collection of DIS compliant data base libraries, specifications and standards. Exercise data bases (including all cell and intercell data bases) draw from the DIS CDB and are constrained by the standards imposed by the DIS CDB.

Components: Models of weapons, sensors, jammers, engines and propulsion systems, etc. which constitute one level in the hierarchy of simulations. Components are generally tightly coupled models which combine to make up platforms. There are no strict rules for defining where components stop and platforms begin, especially for "platforms" such as aircraft carriers which contain other platforms and assemblies of platforms such as air defense batteries. Components can be decomposed into parts or devices. The term component is interchangeable with the term module.

Computer Generated Forces (CGF) Entity: A collection of unmanned battlefield entities under control as a unit. CGF replace or supplement friendly, enemy, or neutral manned simulators during a specific session. If a platform level simulation entity is directly controlled by a man in the loop (whether a participant, OPFOR, or controller), it is considered a manned battlefield entity rather than a CGF entity. The SIMNET program uses the term "semi-automated forces" (SAFOR) for CGF.

DIS cells: Cells containing one or more homogeneous simulation entities; see homogeneous network.

Distributed Interactive Simulation (DIS): 1) A time and space coherent representation of a virtual battlefield environment, measured in terms of human perception and the behaviors of warfighters interacting in free play with other warfighters and/or with computer generated forces. DIS provides a structure by which independently developed systems may interact with each other in a well managed and validated combat simulation environment during all phases of the development process and in subsequent training. 2) The class of simulations defined by the DIS Architecture and associated standards.

Dead Reckoning: A general term used to describe the process of extrapolating platform position based on last known position, velocity, and sometimes higher-order derivatives of position vs. time, and/or other vehicle dynamic characteristics. To avoid confusion between first order dead reckoning (position, velocity) and higher order algorithms, this document uses the term Remote Entity Approximation (REA) for all references to extrapolated positions.

Electronic Battlefield: see Virtual Battlefield.

Entity: A simulation entity.

Environment Entity: The entity responsible for maintaining and disseminating the dynamic information on the state of the geographic, atmospheric, and bathyspheric elements represented in a session. The environment entity is responsible for the broadcast of information concerning changes in the environment including cratering, smoke, building collapse, weather conditions, sea state, etc. regardless of their cause. Its elements correspond to the components of the actual battlefield environment and include terrain (contour, surface, etc.), atmosphere (haze, clouds, wind, etc.), bathysphere (currents, shipping noise, etc.), sun/moon lighting, natural features such as trees and other vegetation, and manmade objects in the environment, such as obstacles, buildings, and bridges. An environment entity has no direct soldier/machine interface for the purpose of control. Thus the environment entity would assume responsibility for a simple mine, but a commanded mine would remain the responsibility of the battlefield entity controlling it.

Exercise: The conduct of a session involving one or more cells over a period of time. The term exercise is used in the same sense that the term is used in the training community. It is equivalent to "test" "experiment", or "study scenario" in other DoD communities.

Exercise Data Base: A name for the union of all cell and intercell data bases in an exercise.

Fidelity: The closeness of the virtual battlefield to the actual battlefield. Fidelity has many parameters including physical fidelity, electromagnetic fidelity, behavioral (for the CGF) fidelity, etc.

Heterogeneous network : A network of DIS objects with partially consistent behaviors and/or partially correlated databases. Examples of heterogeneous networks are networks of simulators of varying fidelity, networks of simulators and actual equipment operating on instrumented ranges, and mixes of simulators and unit level simulations.

Homogeneous network : A network of DIS objects with fully consistent behaviors and fully correlated databases. Normally, this would constitute a single cell, but Cell Interface Units may be used between cells for the purpose of filtering out messages not needed by the other cells such as local tactical communications. For example, this might be done to reduce communications bandwidth requirements for the exercise.

Interoperability: The ability of a set simulation entities to interact with an acceptable degree of fidelity.

Intercell Data Base: The data needed by Cell Interface Units and Cell Adapter Units to support interoperation of cells.

Local Area Network (LAN): A class of data network which provides high data rate interconnection between network nodes in close physical proximity. LANs are defined by the IEEE 802.X series of standards.

Long Haul Networks (LHN): Also called Wide Area Network (WAN). A communications network of devices which are separated by substantial geographical distance. An LHN could be any of numerous networks available commercially or through the government which can accommodate the requirements of the DIS virtual battlefield for long distance network services.

Manned Platform Entity: Corresponds to actual battlefield entities or proposed battlefield entities which are driven, guided, flown, or otherwise have a man or men in the loop. This includes command posts and other C3I nodes and may include role players representing other battlefield entities or staff functions.

Node: 1) Processing node: the hardware and software processing resources devoted to one or more simulation entities. 2) Network node: a specific network address.

Non-standard Cell: A cell which is not compliant with the DIS message and data base standards. Non-standard cells require a Cell Adapter Unit in order to join a DIS exercise.

Physical Realization: The details and mechanics of the underlying networked simulation system which generates the illusion of the virtual battlefield. Physical realization includes both the simulation nodes and the supporting networks.

Platform: A generic term used to describe a level of representation equating to vehicles, fixed sites, individual terrain features, etc. in the hierarchy of representation possibilities. Other representation levels include units (made up of platforms) and components or modules (which make up platforms).

Protocol Data Unit (PDU): A PDU is a structured message which transfers essential data of a specific type from one simulation entity to another and allows them to participate in a common exercise. DIS PDUs comply with the DIS PDU Message Standard.

Remote Entity Approximation (REA): A general term used to describe the process of extrapolating platform position based on last known position, velocity, and sometimes higher-order derivatives of position vs. time, and/or other vehicle dynamic characteristics. The term "dead reckoning" is often used, but dead reckoning implies first-order extrapolation, based on position and velocity. To avoid confusion between first order dead reckoning and higher order algorithms, this document uses the term Remote Entity Approximation (REA) for all references to extrapolated positions.

Semi-Automated Forces (SAFOR): Simulation of friendly, enemy, and neutral platforms on the virtual battlefield in which the individual platform simulations are operated by computer simulation of the platform crew and command hierarchy. The term "semi-automated" implies that the automation is directly controlled and monitored by a human who injects command-level decision making into the automated command process. For the purposes of the DIS architecture, the term Computer Generated Forces (CGF) replaces SAFOR.

Session: A collection of simulation entities configured to interact within a specific virtual battlefield over a given network configuration.

Session Data Base (SESSION): A standard DIS database which includes network initialization data and simulation entity initialization and control data.

Scenario: A scenario describes an exercise in military terms. It is not concerned with the support functions needed to set up, maintain, record, and play back the exercise.

Simulation entity: A generic name for the elements (other than networks and computer interface/adapter units) which comprise a cell. It includes manned and unmanned simulators, simulations, computer generated forces, environment entities, and instrumented operational equipment. A simulation entity can only participate in one exercise at a time.

Simulator: A simulator is a physical representation of an actual battlefield entity, in which the human sensory and control functions of the simulator replicate the human sensory and control functions of the actual battlefield entity.

Simulation: A simulation is a computer replication of actual battlefield entities or collections of entities (units) which are fully automated or partially automated. For the purposes of this document, all simulators are simulations, but not all simulations are simulators. The latter term is reserved for devices where the human interfaces and control functions attempt to replicate those of an actual battlefield device.

Simulation support entity: Processing modules used to support, control, or monitor the simulation environment, but which do not actually exist on the battlefield. This includes the stealth vehicle, the plan view display, After Action Review systems, and simulation control systems.

SIMWORLD: A collection of specifications that define the algorithms and models incorporated in a class of simulation entities. It defines the battlefield terrain modeling algorithms used, atmospheric/bathyspheric models employed, electromagnetic and acoustic spectrums recognized, fidelity characteristics, time reference, supported classes of interactions, etc.

Site: For the purposes of this document, an actual physical location at a specific geographic area, e.g. the Ft. Knox site. A site can contain a single cell, multiple cells, or only part of a cell

Standard Cell: A cell which is compliant with the DIS message and data base standards.

State: The internal status of a simulation entity, e.g. fuel level, number of rounds remaining, location of craters, etc. State messages are used to start and restart entities or to update entities concerning the dynamic changes in the environment in their area of interest.

Stimulator: A stimulator is a battlefield entity consisting of hardware and/or software modules which injects signals directly into the sensor systems of an actual battlefield entity to simulate other battlefield entities in the virtual battlefield. Stimulators also inject simulation messages into the virtual battlefield as necessary for other entities to interact with the stimulator on the virtual battlefield.

Virtual Battlefield: The illusion resulting from simulating the actual battlefield; synonymous with Electronic Battlefield.

Virtual Network: The interconnection of DIS cells by any communications means which provide the necessary network services to conduct a session.

Wide Area Network: see Long Haul Network.

World View: The view each simulation entity maintains of the simulated world from its own vantage point, based on the results of its own simulation and its processing of event messages received from all external entities.