

IEEE Standard for Distributed Interactive Simulation— Application Protocols

Sponsor

**Standards Committee on Interactive Simulation (SCIS)
of the
IEEE Computer Society**

Approved September 21, 1995

IEEE Standards Board

Abstract: Data messages, known as protocol data units (PDUs), that are exchanged on a network between simulation applications are defined. These PDUs are for interactions that take place within specified domains called protocol families, which include Entity Information/Interaction, Warfare, Logistics, Simulation Management, Distributed Emission Regeneration, and Radio Communications.

Keywords: Data messages, protocol data units (PDUs), Distributed Interactive Simulation, Simulation network.

The Institute of Electrical and Electronics Engineers, Inc.
345 East 47th Street, New York, NY 10017-2394, USA

Copyright © 1996 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1996. Printed in the United States of America.

ISBN 1-55937-572-8

No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

IEEE Standards documents are developed within the Technical Committees of the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Board. Members of the committees serve voluntarily and without compensation. They are not necessarily members of the Institute. The standards developed within IEEE represent a consensus of the broad expertise on the subject within the Institute as well as those activities outside of IEEE that have expressed an interest in participating in the development of the standard.

Use of an IEEE Standard is wholly voluntary. The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE Standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE Standard is subjected to review at least every five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE Standard.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of all concerned interests, it is important to ensure that any interpretation has also received the concurrence of a balance of interests. For this reason IEEE and the members of its technical committees are not able to provide an instant response to interpretation requests except in those cases where the matter has previously received formal consideration.

Comments on standards and requests for interpretations should be addressed to:

Secretary, IEEE Standards Board
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
USA

Note: Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying all patents for which a license may be required by an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Authorization to photocopy portions of any individual standard for internal or personal use is granted by the Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; (508) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

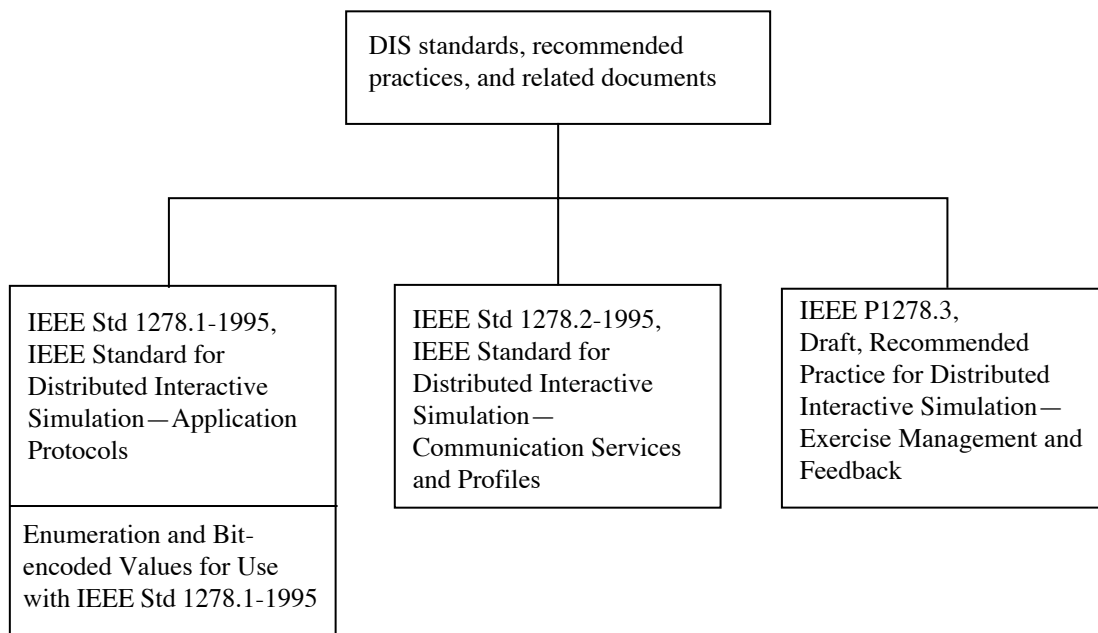
Introduction

(This introduction is not part of IEEE Std 1278.1-1995, IEEE Standard for Distributed Interactive Simulation—Application Protocols.)

Distributed Interactive Simulation (DIS) is a government/industry initiative to define an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual worlds for the simulation of highly interactive activities. This infrastructure brings together systems built for separate purposes, technologies from different eras, products from various vendors, and platforms from various services, and permits them to interoperate. DIS exercises are intended to support a mixture of virtual entities with computer controlled behavior (computer generated forces), virtual entities with live operators (human in-the-loop simulators), live entities (operational platforms and test and evaluation systems), and constructive entities (wargames and other automated simulations). DIS draws heavily on experience derived from the Simulator Networking (SIMNET) program developed by the Advanced Research Projects Agency (ARPA), adopting many of SIMNET's basic concepts and heeding lessons learned.

In order for DIS to take advantage of currently installed and future simulations developed by different organizations, a means had to be found for assuring interoperability between dissimilar simulations. These means were developed in the form of industry consensus standards. The open forum (including government, industry, and academia) chosen for developing these standards was a series of semi-annual Workshops on Standards for the Interoperability of Distributed Simulations that began in 1989. The results of the workshops have been several IEEE standards. These standards provide application protocol and communication service standards to support DIS interoperability. In addition, IEEE P1278.3, Draft, Recommended Practice for Exercise Management and Feedback, provides user guidelines for setting up and conducting a DIS exercise.

The relationship between the component documents comprising the set of IEEE DIS documents is shown in the figure below. Used together, these standards and the recommended practice will help to ensure an interoperable simulated environment.



The interoperability components addressed by these standards and the recommended practice are:

- a) Application protocols
- b) Communications
- c) Exercise management and feedback

IEEE Std 1278.1-1995 defines the format and semantics of data messages, also known as Protocol Data Units (PDUs), that are exchanged between simulation applications and simulation management. The PDUs provide information concerning simulated entity states, the type of entity interactions that take place in a DIS exercise, and data for management and control of a DIS exercise. IEEE Std 1278.1-1995 also specifies the communication services to be used with each of the PDUs.

An additional, non-IEEE document is required for use with IEEE Std 1278.1-1995. This document is entitled *Enumeration and Bit-encoded Values for use with IEEE Std 1278.1*, and is available from the Tactical Warfare Simulation and Technology Information Analysis Center at the Institute for Simulation and Training of the University of Central Florida.

IEEE Std 1278.2-1995 defines the communication services required to support the message exchange described in IEEE Std 1278.1-1995. In addition, IEEE Std 1278.2-1995 provides several communication profiles that meet the specified communications requirements.

Together IEEE Std 1278.1-1995 and IEEE Std 1278.2-1995 provide the necessary information exchange for the communications element of DIS.

P1278.3, currently a draft, provides guidelines for establishing a DIS exercise, managing the exercise, and providing proper feedback. When approved, this recommended practice will be used in conjunction with IEEE Std 1278.1-1995 and IEEE Std 1278.2-1995.

Revisions to each of these standards and the recommended practice are anticipated within the next few years to clarify existing material, to correct possible errors, and to incorporate new related material. Future versions of these documents will contain information concerning additional interoperability components that are currently in the process of being defined.

At the time this standard was completed, the Protocols Working Group had the following membership:

J. Joseph Brann, *Chair*

Emissions

Sam Knight, *Subgroup Chair*

Budd Adams
David Anding
Ronald Bailey
Ray Beaver
David Berkman
Dan Bradford
James Brewer
Eileen Burkhardt
Keith Carlson
Mike Conquest
Brad Cope
Kevin Correll
Claude Crassous de Medeuil
Rudolph Croteau
Larry Cruise
David Daniel
Wim De Jong
David DeFanti
Jerry Denver
Ken Doris
George Downs brough
David Dunmire
Susan Ellison
Don Farquhar
Eric Feathers
Raymond Fidler
Grayden Figart

Jerry Forbes
Darrell Fowler
Richard Gagan
Jim Garrett
Kent Glovanovitz
Robert Greene
David Greschke
C. Edward Hagen
James Harrison
Donald Hauck
Colleen Hogan
Tom Hoog
Charles Howard
Joe Huber
John Huddle
Wallace Hughes
Darren Humphrey
Claus Janota
Dan Katz
John Keane
Jim Kester
Vincent Kijowski
Sam Knight
Pete Latimore
Bob Lucas
Tim Magnani
Cathy Matthews

Mark McAuliffe
David McKeeby
Michael McKeon
Jim Montgomery
Matt O' Connor
Edward Oswald
Allen Ottman
Susan Park
Robert Poling
Joe Primerano
John Rankin
David Roberts
Maurie Roesch
Graham Shanks
Linda Simmons
Claudia Slaton
Brian Smith
Kevin Stever
Michael Streetman
Stephen Swenson
Darrin Taylor
Lorraine Tuliano
Dan Van Hook
Jim Watson
Roger Werner
Keith Wimmer
Douglas Wood

Radio

Richard Schaffer, *Subgroup Chair*

Robert Bassett
John Burnett
Robert Butterfield
Eric R. Christensen
Ken Ciszewski
Daniel Cowan
Claude Crassous de Medeuil
David Daniel
Ann Davies
Nicholas DiPatri
Larry Dworkin
Robert Edge
Susan Ellison
Boyd Fair

Joseph Fann
Allen Farrington
Tom Gehl
Patrick Getchell
Laurence Goldberg
James Gonzalez
C. Edward Hagen
Donald Hauck
Darren Humphrey
Anil Joglekar
Amnon Katz
Martin Katz
Thomas Knapp
John Kutz

Joseph Lacetera
Stephen Mims
Eric Oelschlager
Robert Poling
Robert Rand
David Reed
Richard Schaffer
Steven Specht
Anne Spinosa
Gregory Taylor
Randy Tremblay
Rebecca Turkington
Deborah Wilbert
Douglas Wood

Simulation Management
Michael Robkin, Subgroup Chair

David Berkman
Christina Bouwens
John Brabbs
Kevin Brandt
Ken Ciszewski
Randy Cox
Ann Davies
Grayden Figart
Jerry Forbes
Suresh Goel
Doug Goold
Clark Karr
John Kirkpatrick
Robert Knox

Lee Lacy
Ronald Laird
Norm Lawler
Larry LeDuc
Karl Lindberg
Tom Marlow
Thom McLean
John Mills
Monte Montieroth
Sheila O' Brien
Matthew O' Connor
Barbara Pemberton
David Roberts
Michael Robkin
Marnie Salisbury

Randy Saunders
Richard Schaffer
Graham Shanks
Steven Sheasby
David Shen
Scott Smith
Steven Swaine
Philip Topper
Randall Toth
James Totten
Amy Vanzant-Hodge
Ralph Whitney
Jeff Wicks
Deborah Wilbert

The following persons were on the balloting committee:

Alexander Akerman, III
Robert Bianco
Kevin E. Boner
Christina L. Bouwens
J. Joseph Brann
Don Brutzman
Michael T. Burke
Ernest R. Carbone
John Cater
Frederick Cazenave
Andrew Ceranowicz
Susan Chen
Danny Cohen
Dominic Colonna
Wendell Cook
Claude Crassous de Medeuil
Ann Davies
Robert De Kinder
Jean-Pierre Faye
William F. Flanigan
Jerry Forbes
William Gallip
Keith H. Gardner
Sarah Gautier
Lawrence H. Goldberg
Brian F. Goldiez
Charles E. Hagen
James H. Hammond
Elliott Hinely
Ronald C. Hofer

Mark Hoptiak
Stephen D. Huffman
Michelle Kalphat-Lopez
Mark Kilby
John F. Lethert
Jay Levenson
Robert Levine
Karl Lindberg
Margaret Loper
Gerald Lucha
Michael Macedonia
David L. Macintire
Edward Martin
Ron Matusof
Mark McAuliffe
Peter McCarthy
Larry Bruce McDonald
David R. McKeeby
Duncan C. Miller
Jon T. Miller
John Mills
James R. Moulton
Eugene P. Naccarato
Hung T. Nguyen
Glenn M. O' Bannon
Michael O' Connor
Granville Ott
George Moody Parsons
Dana A. Paterson
Wesley Potterton

J. Mark Pullen
Joe Ricci
Mark Riecken
John Robbins
Michael Robkin
Michael Rothrock
John Ruckstuhl
Randy Saunders
Richard Schaffer
Frank Serna
Steven Sheasby
Jack H. Sheehan
David T. Shen
John E. Sherwood
Steven J. Slosser
Scott H. Smith
Robert Sottolare
Lee Spence
Gerald Stueve
Gary L. Teper
James Totten
Greg J. Unangst
Harold Underwood
Daniel J. Van Hook
Richard Vance
Amy Vanzant-Hodge
Peter Versteegen
Karen Williams
Mark Yeager
Philomena Zimmerman

When the IEEE Standards Board approved this standard on September 21, 1995, it had the following membership:

E. G. “Al” Kiener, *Chair*

Donald C. Loughry, *Vice Chair*

Andrew G. Salem, *Secretary*

Gilles A. Baril
Clyde R. Camp
Joseph A. Cannatelli
Stephen L. Diamond
Harold E. Epstein
Donald C. Fleckenstein
Jay Forster*
Donald N. Heirman
Richard J. Holleman

Jim Isaak
Ben C. Johnson
Sonny Kasturi
Lorraine C. Kevra
Ivor N. Knight
Joseph L. Koepfinger*
D. N. “Jim” Logothetis
L. Bruce McClung

Marco W. Migliaro
Mary Lou Padgett
John W. Pope
Arthur K. Reilly
Gary S. Robinson
Ingo Rüschi
Chee Kiow Tan
Leonard L. Tripp
Howard L. Wolfman

*Member Emeritus

Also included are the following nonvoting IEEE Standards Board liaisons:

Satish K. Aggarwal
Richard B. Engelman
Robert E. Hebner
Chester C. Taylor

Kim Breitfelder
IEEE Standards Project Editor

Contents

CLAUSE	PAGE
1. Overview.....	1
1.1 Scope.....	1
1.2 Terminology.....	1
1.3 Key concepts.....	2
2. References.....	7
3. Definitions, acronyms, and abbreviations.....	8
3.1 Definitions.....	8
3.2 Acronyms and abbreviations.....	11
4. General requirements.....	11
4.1 DIS exercise.....	11
4.2 Issuing entity.....	11
4.3 Issuance of PDUs.....	12
4.4 Receipt of PDUs.....	12
4.5 PDUs for DIS.....	12
5. Detailed requirements.....	53
5.1 Representation of data.....	53
5.2 Basic data types and records.....	55
5.3 PDUs for DIS.....	74
ANNEX	
Annex A (normative) Articulated and attached parts.....	119
Annex B (normative) Dead reckoning definitions and algorithms.....	123
Annex C (informative) Bibliography.....	130

IEEE Standard for Distributed Interactive Simulation— Application Protocols

1. Overview

This standard explains the information technology protocols required for Distributed Interactive Simulation (DIS) applications. This standard is divided into five clauses. Clause 1 provides the scope of the standard, and details key DIS concepts that will help in understanding the context of this standard. Clause 2 lists references to other standards that are useful in applying this standard. Clause 3 provides definitions of terms and acronyms that are used in the standard. It is imperative for the user of this standard to thoroughly review these definitions before proceeding on to the other clauses. Clause 4 contains requirements concerning the content and use of protocol data units (PDUs) in DIS exercises. Clause 5 defines the various PDUs and their fields.

1.1 Scope

This standard is part of a proposed set of standards for DIS applications. Each standard in the proposed set describes one or more of the elements that constitute the DIS environment. As a whole, the set of standards will define an interoperable simulated battle environment. This particular standard addresses the application protocols.

This standard defines the data messages, known as PDUs, that are exchanged on a network between simulation applications. The messages defined are for interactions that take place within specified domains called protocol families. Current protocol families include Entity Information/Interaction, Warfare, Logistics, Simulation Management, Distributed Emission Regeneration, and Radio Communications. Future versions of this standard will contain additional PDUs and protocol families to exchange information about interactions and functions not currently supported.

1.2 Terminology

The following terms, which are defined in clause 3, form the basis for understanding the key concepts stated in 1.3.

- a) Distributed Interactive Simulation (DIS)
- b) Host computer
- c) Simulation application
- d) Simulation entity
- e) Simulation exercise
- f) Simulation environment

1.3 Key concepts

This subclause contains details on key DIS concepts that will help the reader understand the context of this standard. This information is of a general or explanatory nature that may be helpful, but is not mandatory.

1.3.1 Basic architecture concepts

The basic architecture concepts of DIS are an extension of the Simulator Networking (SIMNET) program developed by the Advanced Research Project Agency (ARPA). The basic architecture concepts for DIS, and their implications as they apply to DIS, are:

- a) *No central computer controls the entire simulation exercise.* Some simulation systems have a central computer that maintains the world state and calculates the effects of each entity's actions on other entities and the environment. These computer systems must be sized with resources to handle the worst-case load for a maximum number of simulated entities. DIS uses a distributed simulation approach in which the responsibility for simulating the state of each entity rests with separate simulation applications residing in host computers connected via a network. As new host computers are added to the network, each new host computer brings its own resources.
- b) *Autonomous simulation applications are responsible for maintaining the state of one or more simulation entities.* Simulation applications (or simulations) are autonomous and generally responsible for maintaining the state of at least one entity. In some cases, a simulation application will be responsible for maintaining the state of several entities. As the user operates controls in the simulated or actual equipment, the simulation is responsible for modeling the resulting actions of the entity using a simulation model. That simulation is responsible for sending messages to others, as necessary, to inform them of any observable actions. All simulations are responsible for interpreting and responding to messages of interest from other simulations and maintaining a model of the state of entities represented in the simulation exercise. Simulations may also maintain a model of the state of the environment and nondynamic entities, such as bridges and buildings, that may be intact or destroyed.
- c) *A standard protocol is used for communicating ground truth data.* Each simulation application communicates the state (which is herein called ground truth) of the entity it controls/measures (location, orientation, velocity, articulated parts position, etc.) to other simulations on the network. The receiving simulation is responsible for taking this ground truth data and calculating whether the entity represented by the sending simulation is detectable by visual or electronic means. This perceived state of the entity is then displayed to the user as required by the individual simulation.
- d) Changes in the state of an entity are communicated by its controlling simulation application.
- e) Perception of events or other entities is determined by the receiving application.
- f) *Dead reckoning algorithms are used to reduce communications processing.* A method of position/orientation estimation, called dead reckoning, is used to limit the rate at which simulations must issue state updates for an entity. Each simulation maintains an internal model of the entity it represents. In addition, the simulation maintains a dead reckoning model of its entity. The dead reckoning model represents the view of that entity by other simulation applications on the network and is an extrapolation of position and orientation state using a specified dead reckoning algorithm. On a regular basis, the simulation compares the internal model of its entity to the dead reckoning model of the entity. If the difference between the two exceeds a predetermined threshold, the simulation will update the dead reckoning model using the information from the internal model. At the same time, the simulation will send updated information to other simulations on the network so that they can update their dead reckoning model of the entity. By using dead reckoning, simulations are not required to report the status of their entities as often.

1.3.2 Coordinate systems

Locations in the simulated world are identified using a right-handed, geocentric Cartesian coordinate system called the world coordinate system. The shape of the world is described in DMA TR 8350.2.¹ The origin of the coordinate system is the centroid of the earth. The axes of this system are labeled X , Y , and Z , with the positive X -axis passing through the prime meridian at the equator, with the positive Y -axis passing through 90° east longitude at the Equator and the positive Z -axis passing through the north pole as shown in figure 1. A distance of one unit measured in world coordinates corresponds to a distance of 1 m in the simulated world. A straight line in the world coordinate system is a straight line in the simulated world. This is a rotating reference frame that rotates on a daily period as the earth rotates.

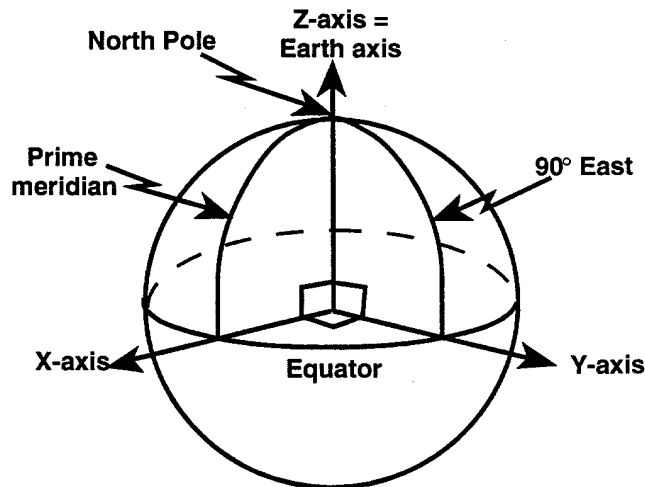


Figure 1—World coordinate system

To describe the location and orientation of any particular entity, an entity coordinate system is associated with the entity. This is also a right-handed Cartesian coordinate system with the distance of one unit corresponding to 1 m as in the world coordinate system. The origin of the entity coordinate system is the center of the entity's bounding volume excluding its articulated and attached parts. The axes are labeled x , y , and z with the positive x -axis pointing to the front of the entity, the positive y -axis pointing to the right side of the entity, and the positive z -axis pointing out the bottom of the entity as shown in figure 2.

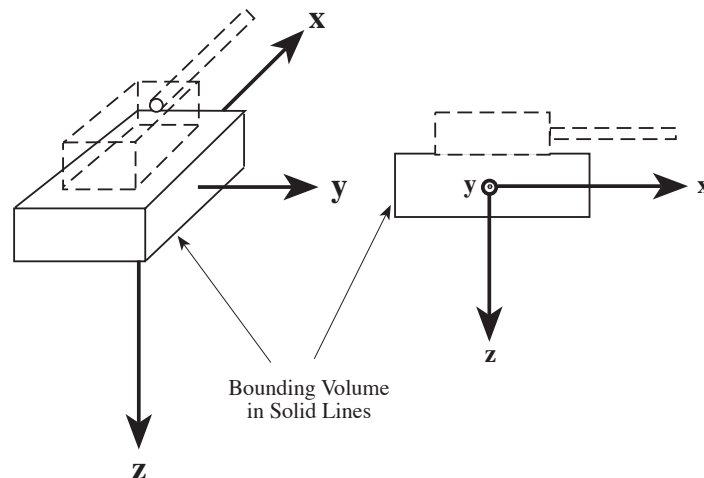


Figure 2—Entity coordinate system

¹Information on references can be found in clause 2.

The location of an entity is specified as the position of the origin of its entity coordinate system, expressed in world coordinates. The entity's orientation is specified using three angles that describe the successive rotations needed to transform from the world coordinate system to the entity coordinate system. These angles are called Euler angles and specify a set of three successive rotations about three different orthogonal axes as shown in figure 3. The order of rotation is first, rotate about z by the angle ψ (ψ), then about the new y (y') by angle θ (θ), then about the newest x (x'') by the angle ϕ (ϕ). The positive direction of rotation about an axis is defined as clockwise when viewed toward the positive direction along the axis of rotation.

1.3.3 Communication services

The communication services required by each DIS PDU are described in detail in IEEE Std 1278.2-1995.

1.3.4 Functional requirements for DIS

DIS is intended to support the following functional requirements:

- a) Entity information/interaction
- b) Warfare
- c) Logistics
- d) Radio communications
- e) Distributed emission regeneration
- f) Management
- g) Environment information

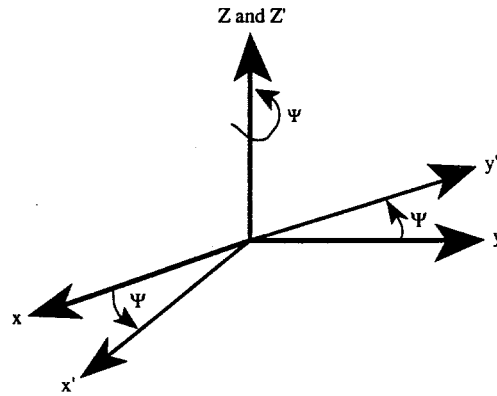
Currently requirements a) through f) are supported by this standard. Enhancement of these areas and the addition of capabilities for area g) are the subject of future standards. A brief description of each is given in 1.3.4.1 and 1.3.4.2.

1.3.4.1 Current capabilities

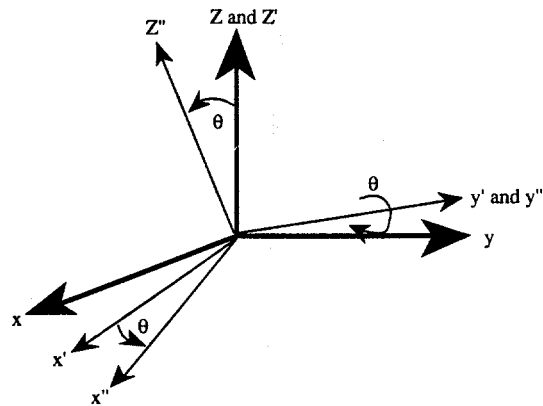
The entity information exchanged between simulation applications includes the type of entity, its location, its orientation, and how the entity might appear to others. The simulation entity could be a vehicle, a building, a munition (such as a missile), or a cloud of smoke. DIS requires that entities be enumerated based on their entity type, allowing a variety of different entities to be represented. Sending the location and orientation of an entity is critical for correct representation of the entity by other simulations on the network. Inclusion of the velocity and the acceleration parameters allows receiving simulations to employ higher-level, higher-accuracy extrapolation routines. See 1.3.1 item f). The visual appearance of an entity can be expressed in a number of ways. Entities may be on fire or smoking, or an entity may emit engine smoke or have a wake trailing behind in the water, or an entity may have entity-specific lights actuated, all of which affect the visual appearance of the entity.

Throughout a simulation exercise, the state information associated with the interactions that take place between entities needs to be exchanged. Interactions that are currently supported include collisions. In the event that two entities collide, the simulations controlling the entities must be informed of the collision. A message about the collision is sent by each simulation application when it detects that its entity has collided with another entity. Each simulation application determines the damage to its own entity based on information in the collision message.

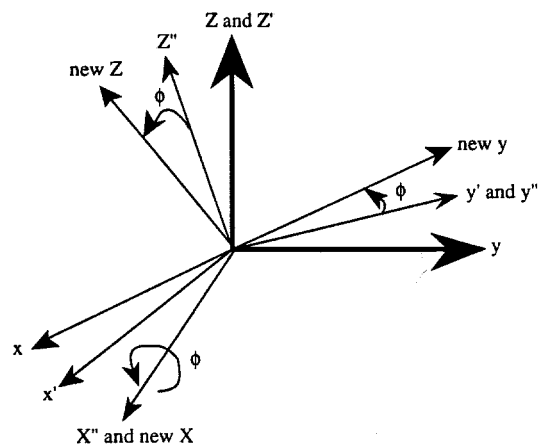
Warfare in a DIS exercise involves the firing and detonation of munitions. When an entity fires a weapon, the simulation application controlling the entity needs to communicate the location of the firing weapon and the type of munition fired. The detonation of the munition is also communicated by the simulation application controlling the munition. Using the information in the detonation message, all simulation applications controlling affected entities assess damage to their entities.



a) First, rotate about z by angle ψ (ψ)



b) Second, rotate about y (y') by θ (θ)



c) Third, rotate about newest x (x'') by ϕ (ϕ)

Figure 3—Definition of Euler angles

Repair and resupply logistic services are modeled in as simulation exercise by means of the Logistics family of PDUs. Messages representing requests for services and the transfer of supplies are exchanged between simulation applications that are providers of the repair or resupply service, and those simulated entities in need of such service.

Audio and digital message communications play an important role in certain types of DIS exercises. The sending (transmitting) entity sends a message defining the details of the communicating device, and then the communicated message (data). Entities receiving the message can determine their capability to receive the transmitted data, and subsequently how to process the received data.

Representation of lasers, active electromagnetic emissions, and acoustic emissions including active counter-measures are essential in certain simulation exercises. Emitting entities simulate their emitter and output real-time operational parameters. Receiving entities can then regenerate the transmitted signal based upon the simulated emitter output data and stored database. Each receiving entity is responsible for determining if the emission is detectable. If detectable, the receiving entity will use the emission data to appropriately influence its detection equipment or simulation of that equipment.

Management of a simulation exercise is desirable to facilitate the operation of the network and certain aspects of the simulation exercise. DIS management functions can be divided into network management and simulation management. Network management functions handle the basic network functions such as load management, monitoring of nodes and gateways, and error recovery. The network manager would also have knowledge of host computers on the network, such as their physical locations and network addresses. Analysis of network performance would be performed here. There is also a need for management of the overall simulation exercise. Functions of simulation management include: starting, restarting, pausing, and stopping an exercise; the creation and removal of entities from an exercise; and collection and distribution of data with simulation applications.

1.3.4.2 Future capabilities

A number of capabilities that are not part of this standard are expected to be in future versions of this standard or in related documents.

In addition to visual appearance, an entity may also have a certain infrared signature, or an entity can radiate an acoustic signature that is unique to the entity. Consequently, each entity has several appearances. These include emission of sound, light, and heat, as well as the reflection of light, radar, and sound (sonar).

For simulation entities to participate in the same exercise, they must have access to the same environment information. Different types of information about the environment are necessary to make the exercise as realistic as possible. This information may include changes in the terrain, weather, and ambient illumination. Changes in the terrain can be caused by a number of factors. These include engineering effects such as the construction of a bridge, berm, or a building; weapons effects that could destroy objects created via engineering effects, as well as change the shape of the terrain due to the impact of shells or explosion of mines; or natural effects such as flooding. Conditions such as rain, snow, fog, or clouds need to be represented in DIS to add to the realism of a DIS exercise. The wind and its effect on a cloud of smoke can affect vehicle visibility. Chemical clouds and their dispersal can affect dismounted infantry. Atmospheric conditions can affect radio communications and radar performance between entities. Night battles, as well as day battles, should be simulated. DIS will support battles that can take place in any degree of ambient illumination. The subsurface environment plays a significant role in simulations employing acoustic and seismic sensors and in simulations containing subsurface entities (land mines, submarines, etc.). Oceanographic and geophysical representations will use correlated environment data to simulate these effects where appropriate.

Application protocols supporting field instrumentation applications associated with instrumented live participants are not fully addressed in this standard. Requirements for application protocol suitable for use with real entities on instrumented ranges operating within the DIS environment and interacting with

simulated and real entities will be addressed in future versions of this family of DIS standards. Limited communication bandwidth, volume of data, methods of data transmission, and type of data transmitted are among the issues to be addressed.

It may not be necessary that all participants receive all PDUs of an exercise. Mechanisms to allow for efficient use of network resources may include filtering PDUs at various locations in the network, data compression schemes, and sending only changes in the PDU information rather than the entire PDU. Multicast addressing schemes and protocols will be used to create and control groups where PDUs are only sent to members of that group. The multicast mechanisms are currently being developed and will be specified in the areas of communication architecture, network management, and simulation management.

2. References

This standard shall be used in conjunction with the following publications. When the following standards are superseded by an approved revision, the revision shall apply.

DMA TR 8350.2, 1987, Department of Defense World Geodetic System 1984 (WGS 84), Its Definition and Relationships with Local Geodetic Systems.²

IEEE Std 754-1985 (Reaff 1990), IEEE Standard for Binary Floating-Point Arithmetic (ANSI).³

IEEE Std 1278.2-1995, IEEE Standard for Distributed Interactive Simulation—Communication Services and Profiles.

IEEE P1278.3 (Draft), Recommended Practice for Distributed Interactive Simulation—Exercise Management and Feedback, June 1990.⁴

IST-CR-94-37, Standard for Distributed Interactive Simulation—Fidelity Description Requirements.⁵

IST-CR-95-14, Enumeration and Bit-encoded Values for use with IEEE Std 1278.1-1995, Standard for Distributed Interactive Simulation—Application Protocols; minimum compliant version is IST-CR-95-05 with change request 344.

ISO 7498: 1984, Information processing systems—Open Systems Interconnection—Basic Reference Model.⁶

ITU Recommendation G711, “Pulse Code Modulation (PCM) of Voice Frequencies,” in CCITT Blue Book Volume III Fascicle 4, General aspects of digital transmission systems: terminal equipments, 1988.⁷

MIL-STD-1821, Military Standard Simulator Data Base (SSDB) Interchange Format (SIF) Design Standard.⁸

²This publication is available from National Technical Information Service (NTIS), 5285 Port Royal Road Springfield, VA 22161, USA.

³IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

⁴This IEEE standards project was not approved by the IEEE Standards Board at the time this publication went to press. For information about obtaining a draft, contact the IEEE.

⁵IST documents are available from the University of Central Florida, Center for Continuing Education, Orlando, FL, 32816-0177, USA.

⁶ISO publications are available from the ISO Central Secretariat, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse. ISO publications are also available in the United States from the Sales Department, American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036, USA.

⁷ITU publications are available from the International Telecommunications Union, Sales Section, Place des Nations, CH-1211, Genève 20, Switzerland/Suisse. They are also available in the United States from the U.S. Department of Commerce, Technology Administration, National Technical Information Service (NTIS), Springfield, VA 22161, USA.

⁸MIL publications are available from Customer Service, Defense Printing Service, 700 Robbins Ave., Bldg. 4D, Philadelphia, PA 19111-5094.

SI, Le Système International d'Unités: See for example:

- a) "The Complete Metric System with the International System of Units, (SI)," A.L. LeMaraic and John P. Ciaramella, Abbey Books, 1973.
- b) "Guide for the Use of the International System of Units: The Modernized Metric System," A.O. McCoubrey, Sept. 91, NIST/SP-811. Also available from Supt. of Docs. as SN003-003-03113-5.
- c) "Interpretation of the SI for the United States and Metric Conversion Policy for Federal Agencies," B.N. Taylor, Oct. 91, NIST/SP814. Also available from Supt. of Docs.

SIMNET, BBN RPT. 7627—The SIMNET Network and Protocols.⁹

3. Definitions, acronyms, and abbreviations

This clause contains definitions, acronyms, and abbreviations as they are used in this standard.

3.1 Definitions

3.1.1 application layer (Layer 7): The layer of the OSI reference model (ISO 7498: 1984) that provides the means for simulation applications to access and use the network's communications resources.

3.1.2 articulated part: A visible part of a simulated entity that may not move relative to the entity, but is able to move relative to the entity itself.

3.1.3 attached part: A visible part of a simulated entity that may not move relative to the entity, but that may or may not be present. For example, a bomb on an aircraft wing station.

3.1.4 ballistic munition: Any munition that follows a ballistic trajectory.

3.1.5 best effort service: A communication service in which transmitted data is not acknowledged. Such data typically arrives in order, complete and without errors. However, if an error occurs, or a packet is not delivered, nothing is done to correct it (e.g., there is no retransmission).

3.1.6 bit: The smallest unit of information in the binary system of notation.

3.1.7 bounding volume: The six-sided, rectangular enclosing space whose width, length and height are aligned with those of the entity.

3.1.8 dead reckoning: A method for the estimation of the position/orientation of an entity based on a previously known position/orientation and estimates of time and motion.

3.1.9 distributed interactive simulation (DIS): A time and space coherent synthetic representation of world environments designed for linking the interactive, free-play activities of people in operational exercises. The synthetic environment is created through real-time exchange of data units between distributed, computationally autonomous simulation applications in the form of simulations, simulators, and instrumented equipment interconnected through standard computer communicative services. The computational simulation entities may be present in one location or may be distributed geographically.

3.1.10 emitter: A device that is able to discharge detectable electromagnetic, seismic, or acoustic energy.

3.1.11 entity: *See:* **simulation entity.**

⁹See footnote 2.

3.1.12 entity coordinate system: A system whereby location with respect to a simulation entity is described by an entity coordinate system. See 1.3.2

3.1.13 Euler angles: A set of three angles used to describe the orientation of an entity as a set of three successive rotations about three different orthogonal axes (x , y , and z). The order of rotation is first about z by angle ψ (ψ), then about the new y by angle θ (θ), then about the newest x by angle ϕ (ϕ) (see figure 3). Angles ψ and ϕ range between $\pm\pi$, while angle θ ranges only between $\pm\pi/2$ radians. These angles specify the successive rotations needed to transform from the world coordinate system to the entity coordinate system. The positive direction of rotation about an axis is defined by the right-hand rule.

3.1.14 exercise: *See: simulation exercise.*

3.1.15 fidelity: The degree to which the representation within a simulation is similar to a real-world object, feature, or condition in a measurable or perceivable manner.

3.1.16 field: A series of contiguous bits treated as an instance of a particular data type that may be part of a higher level data structure.

3.1.17 guise: A function that provides the capability for an entity to be viewed with one appearance by one group of participants, and with another appearance by another group.

3.1.18 host computer: A computer that supports one or more simulation applications. All host computers participating in a simulation exercise are connected by network(s) including local area networks, wide area networks, radio frequency links, etc.

3.1.19 multicast: A transmission mode in which a single message is sent to multiple network destinations, (i.e., one-to-many).

3.1.20 network management: The collection of administrative structures, policies, and procedures that collectively provide for the management of the organization and operation of the network as a whole.

3.1.21 node: A general term denoting either a switching element in a network or a host computer attached to a network.

3.1.22 octet: A sequence of eight bits, usually operated upon as a unit.

3.1.23 protocol: A set of rules and formats (semantic and syntactic) that determines the communication behavior of simulation applications.

3.1.24 protocol data unit (PDU): A DIS data message that is passed on a network between simulation applications according to a defined protocol.

3.1.25 real-world time: The actual time in the real world, expressed as Universal Coordinated Time (UTC).

3.1.26 reliable service: A communication service in which the received data is guaranteed to be exactly as transmitted.

3.1.27 right-hand rule: Positive rotation is clockwise when viewed toward the positive direction along the axis of rotation.

3.1.28 simulation application: The executing software on a host computer that models all or part of the representation of one or more simulation entities. The simulation application represents or simulates real-world phenomena for the purpose of training or experimentation. Examples of simulation applications include manned vehicle simulators, computer generated forces, environment simulators and computer

interfaces between a DIS network and real equipment. The simulation application receives and processes information concerning entities created by peer simulation applications through the exchange of DIS PDUs. More than one simulation application may simultaneously execute on a host computer. The simulation application is the application layer protocol entity that implements the protocol defined in this standard.

NOTE—This document will use the term *simulation application* to avoid confusion between protocol entities and simulation entities. The term *simulation* may also be used in place of simulation application.

3.1.29 simulation entity: An element of the synthetic environment that is created and controlled by a simulation application and affected by the exchange of DIS PDUs. Examples of types of simulated entities are: tank, submarine, carrier, fighter aircraft, missiles, bridges, or other elements of the synthetic environment. It is possible that a simulation application may be controlling more than one simulation entity.

NOTE—Simulation entities may also be referred to as *entities* in this standard.

3.1.30 simulation environment: The operational environment surrounding the simulation entities. This environment includes terrain, atmospheric, and oceanographic information. It is assumed that participants in the same DIS exercise will be using environment information that is adequately correlated for the type of exercise to be performed.

3.1.31 simulation exercise: An exercise that consists of one or more interacting simulation applications. Simulations participating in the same simulation exercise share a common identifying number called the exercise identifier. These simulations also utilize correlated representations of the synthetic environment in which they operate.

3.1.32 simulation management: A process that provides centralized control of the simulation exercise. Functions of simulation management include: start, restart, maintenance, shutdown of the exercise, and collection and distribution of certain types of data.

3.1.33 simulation time: The reference time (e.g., UTC) within a simulation exercise. This time is established ahead of time by the simulation management function and is common to all participants in a particular exercise.

3.1.34 synthetic environment: The integrated set of data elements that define the environment within which a given simulation application operates. The data elements include information about the initial and subsequent states of the terrain including cultural features, atmospheric and oceanographic environments throughout a DIS exercise. The data elements include databases of externally observable information about instantiable DIS entities and are adequately correlated for the type of exercise to be performed.

3.1.35 tracked munition: A munition for which tracking data is required. By necessity, a tracked munition becomes a simulation entity during its flight; its flight path is represented, therefore, by Entity State PDUs.

3.1.36 unicast: A transmission mode in which a single message is sent to a single network destination, (i.e., one-to-one).

3.1.37 world coordinate system: The right-handed geocentric Cartesian system. The shape of the world is described in DMA TR 8350.2, 1987. The origin of the world coordinate system is the centroid of the earth. The axes of this system are labeled X, Y, and Z, with the positive X-axis passing through the prime meridian at the equator, with the positive Y-axis passing through 90° east longitude at the equator and the positive Z-axis passing through the north pole.

3.2 Acronyms and abbreviations

The following acronyms and abbreviations are used in this standard.

DER	Distributed Emission Regeneration
DIS	Distributed Interactive Simulation
DRM	Dead Reckoning Model
EBV-DOC	An abbreviation for the document, <i>Enumeration and Bit-Encoded Values for Use with IEEE Std 1278.1-1995</i> . This document is frequently updated.
EW	Electronic Warfare
ID	Identification
PDU	Protocol Data Unit
RCP	Radio Communications Protocol
SA	Simulation Application (used only in figures)
SI	The official abbreviation for “Le Système International d’Unités”: A universal system of metric weights and measures adopted in 1960 by the international authority on the metric system, the Conférence Générale de Poids et Mesures (CGPM).
SM	Simulation Manager
TDL	Tactical Data Link
UTC	Universal Coordinated Time

4. General requirements

This clause contains requirements concerning the content and use of PDUs in DIS exercises. All numeric values specified in this standard and its annexes are listed in base ten unless otherwise noted by the suffix (H), which denotes a hexadecimal number. Symbolic names written in all capital letters are used in this standard. The values assigned to the symbolic names are given in clause 5.

4.1 DIS exercise

A DIS exercise shall consist of two or more interacting simulation applications. The DIS PDUs issued by all simulation applications participating in the same exercise shall share one identifying number called the exercise identifier.

4.2 Issuing entity

In a DIS exercise, simulation applications may represent one or more entities. The simulation application shall issue PDUs for each of the entities that it simulates. PDUs shall never be issued in advance of the occurrence of the event or state communicated by the PDU. For reference, it is easier to state that the entity

is issuing the PDU, rather than stating that the simulation application that is simulating the entity is issuing the PDU. Therefore, the phrase *entity issues* shall mean *the simulation application representing the entity issues*.

4.3 Issuance of PDUs

PDUs shall be issued according to the requirements specified in 4.5.

4.4 Receipt of PDUs

Upon receipt of PDUs, simulation applications shall act as specified in 4.5. Unless otherwise stated, the actions described shall apply to all the simulation entities to which the PDU is addressed. Depending on the PDU type, the PDU can be addressed to a particular entity or to all entities within an exercise. Simulation applications shall accommodate out-of-order delivery of PDUs.

4.5 PDUs for DIS

The content and the procedure for use of PDUs in a DIS exercise are established in this subclause.

4.5.1 PDU header

A PDU header shall be the first part of each PDU. The header shall specify the identification number associated with the DIS exercise, the protocol version, the type code for PDU, the protocol family to which the PDU belongs, a timestamp, and the length of the PDU (see also 5.2.24).

NOTE—The requirement to place the PDU header as the first part of each PDU is a global requirement for all PDUs, and thus the PDU header is not listed in the subclauses that identify the information contained in each specific PDU.

4.5.1.1 Protocol version

The PDU header shall specify the version of DIS protocol to which the PDU pertains.

4.5.1.2 DIS exercise identification

Each DIS exercise shall be distinguished from other exercises by the use of an exercise identifier. An identifier that is currently not in use on the network shall be assigned. The management and assignment of exercise identifiers to an exercise is outside the scope of this standard.

4.5.1.3 Types and families of PDUs

This standard defines the following 27 PDUs, which are organized into six protocol families:

- a) Entity Information/Interaction
 - 1) Entity State PDU
 - 2) Collision PDU
- b) Warfare
 - 1) Fire PDU
 - 2) Detonation PDU
- c) Logistics
 - 1) Service Request PDU
 - 2) Resupply Offer PDU
 - 3) Resupply Received PDU
 - 4) Resupply Cancel PDU

- 5) Repair Complete PDU
- 6) Repair Response PDU
- d) Simulation Management
 - 1) Start/Resume PDU
 - 2) Stop/Freeze PDU
 - 3) Acknowledge PDU
 - 4) Action Request PDU
 - 5) Action Response PDU
 - 6) Data Query PDU
 - 7) Set Data PDU
 - 8) Data PDU
 - 9) Event Report PDU
 - 10) Comment PDU
 - 11) Create Entity PDU
 - 12) Remove Entity PDU
- e) Distributed Emission Regeneration
 - 1) Electromagnetic Emission PDU
 - 2) Designator PDU
- f) Radio Communications
 - 1) Transmitter PDU
 - 2) Signal PDU
 - 3) Receiver PDU

In addition to these PDUs, other PDUs and protocol families may be added to this standard in the future. PDU types and protocol families have been reserved for growth. PDU types and protocol families in the range of 129 through 255 have been reserved for experimental purposes.

4.5.1.4 Timestamp

The Timestamp field shall specify the time at which the data in the PDU is valid.

4.5.1.5 Length of PDU

The Length field shall specify the number of octets from and including the first octet of the header to and including the last octet of the PDU.

4.5.2 Entity information/interaction

Information associated with the appearance and location of an entity shall be communicated in a DIS exercise through the use of an Entity State PDU (see also 5.3.3.1). Information associated with collisions between entities shall be communicated in a DIS exercise through the use of the Collision PDU (see also 5.3.3.2).

4.5.2.1 Entity State PDU

The Entity State PDU shall communicate information about an entity's state. This includes state information that is necessary for the receiving simulation applications to represent the issuing entity in the simulation applications' own simulation.

4.5.2.1.1 Information contained in the Entity State PDU

The Entity State PDU shall contain the following information:

- a) Identification of the entity that issued the PDU
- b) Identification of the force to which the entity belongs

- c) Issuing entity's specific entity type
- d) Issuing entity's alternate entity type for use with the Guise function (see 4.5.2.1.5)
- e) Information about the location of the entity in the simulated world and its orientation, including
 - 1) Location with respect to the world
 - 2) Velocity
 - 3) Orientation
 - 4) Dead reckoning parameters that should be employed when extrapolating the position of this entity (Values in this field shall include dead reckoning algorithm in use, linear acceleration, and angular velocity. Other values for this field are as defined in annex B.)
- f) Information required for representation of the entity's appearance including
 - 1) Appearance of the entity (e. g., normal, smoking, on fire, producing a dust cloud, etc.)
 - 2) Markings
 - 3) Number of articulation parameters and the parameter values to represent orientation of articulated parts
 - 4) Presence (including types and numbers) of attached parts or stores
- g) Capabilities of the entity, including
 - 1) Resupply
 - 2) Repair

4.5.2.1.2 Dead reckoning

A method of position/orientation estimation called dead reckoning shall be employed to limit the rate at which Entity State PDUs are issued.

4.5.2.1.2.1 Dead reckoning and the issuing entity

Each simulation application shall maintain two state models of each entity it is representing in support of the dead reckoning process. One model shall be the internal model used by the simulation application to represent its entity. The other model shall be a dead reckoning model of the entity it is representing. Certain thresholds shall be established as criteria for determining if the entity's actual position/orientation has varied an allowable amount from its dead reckoned position/orientation. When the entity's actual position/orientation has varied from the dead reckoned position/orientation by more than a threshold value, the simulation application shall issue an Entity State PDU to communicate the entity's actual position and orientation to other simulation applications. The simulation application shall also use the same information communicated to other simulation applications to update its dead reckoning model of its entity. The default method of dead reckoning error calculation shall be a threshold change in any direction or orientation. The default position threshold shall be DRA_POS_THRSH_DFLT, and the default orientation shall be DRA_ORIENT_THRSH_DFLT.

4.5.2.1.2.2 Dead reckoning and the receiving entity

Each simulation application shall maintain a dead reckoning model of the position/orientation of entities that are of interest (within sight or range) as specified by the dead reckoning model in use. The dead reckoned position/orientation of other entities shall be used by the simulation's application as required for its execution. When the simulation application receives a new update from one of the entities it is dead reckoning, it shall correct its dead reckoning model and base its estimations on the most recent position/orientation, velocity, and acceleration information. Smoothing techniques may be used to eliminate jumps that occur in a visual display when the dead reckoning position/orientation of an entity is corrected using more recent position/orientation data.

4.5.2.1.2.3 Dead reckoning algorithms

Required dead reckoning algorithms for use with this standard are included in annex B. To allow for parameters associated with the dead reckoning algorithm in use, a field has been set aside in the Entity State PDU for dead reckoning parameters.

4.5.2.1.2.4 Dead reckoning of frozen entities

An entity with the Appearance field Frozen Status bit set to Frozen is frozen. A simulation that receives an Entity State PDU from a frozen entity shall not continue dead reckoning extrapolation for that entity.

4.5.2.1.3 Issuance of the Entity State PDU

A simulation shall issue an Entity State PDU when any of the following occur:

- a) Discrepancy between an entity's actual state (as determined by its own internal model) and its dead reckoned state (state using specified dead reckoning algorithms) exceeds a predetermined threshold (see annex B concerning dead reckoning, and 4.5.2.1.2.1 concerning threshold values). This threshold includes changes in position/orientation information and articulated part parameter information (see A.2.1.6).
- b) Change in the entity's appearance occurs. This change may include beginning to burn or smoke.
- c) A predetermined length of real-world time has elapsed since the issuing of the last Entity State PDU. This value and associated tolerance may be established at the start of the exercise or during the exercise by means of Simulation Management PDUs. If no value is established, the default value shall be HRT_BEAT_TIMER. A default tolerance of $\pm 10\%$ shall apply to this value.
- d) Entity ceases to exist in the synthetic environment. A final Entity State PDU shall be issued to indicate that the entity is in the Deactivated state.
- e) Change in the entity's capabilities.
- f) Change in the dead reckoning algorithm used by the simulation application.

The Entity State PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

4.5.2.1.4 Receipt of the Entity State PDU

Upon receipt of the Entity State PDU, a simulation application shall determine whether the PDU contains more current information than that currently being used to model the transmitting entity. If so, the simulation application shall use the information contained therein to model the position, orientation, and appearance of the entity that issued the PDU. Otherwise the PDU shall be discarded.

If the appearance of an entity indicates it is in the Deactivated state or a predetermined length of time has elapsed since any entity's last Entity State PDU, then all simulations shall remove that entity from the exercise. The length of time shall be established at exercise start, although it may be changed during the exercise. If no value is established, the default value for this time shall be the value of the Entity State PDU issuance interval [see 4.5.2.1.3 item c)] multiplied by HRT_BEAT_MPLIER.

4.5.2.1.5 Guise function

The purpose of the Guise function is to allow DIS participants to serve in the opposing force role while operating friendly simulations. This function allows both sides of an engagement to see their team members

as friendly and the opposing forces as hostile. The Guise function only affects the depiction of the other force, not any of the internal characteristics of its simulation or behavior.

The Entity Type field shall be used by members of the issuing entity's team to display the issuing entity. The Alternate Entity Type field shall be used by members of all other teams to display the issuing entity. The force identification determines the team membership of the issuing entity.

NOTE—The location in the Entity State PDU is the origin of the Entity Type model, not the origin of the Alternate Entity Type model. For vehicles in contact with the ground, it will generally be necessary to shift the model for the Alternate Entity Type entity in order to keep it in contact with the ground. This shift is required because the height of the center of the bounding volumes of the Entity Type and Alternate Entity Type models will generally be different.

4.5.2.1.5.1 Issuing entity actions

If the Guise function is used, an issuing entity shall perform the following actions:

- a) Entity shall assign the entity type code for its friendly force appearance to the Entity Type field of the Entity State PDUs it issues
- b) Entity shall assign the entity type code for its opposing force appearance to the Alternate Entity Type field of the Entity State PDUs it issues
- c) In the case of neutral entities, the same type code shall be assigned to both the Entity Type field and the Alternate Entity Type field

If the Guise function is not used, the same entity type shall be assigned to both the Entity Type field and the Alternate Entity Type field.

4.5.2.1.5.2 Receiving entity actions

The receiving entity shall interpret the Entity Type field and Alternative Entity Type field. Upon receiving an Entity State PDU from an entity that is of interest, the receiving entity shall check the value of the force identification of the entity that issued the PDU. The following conditions apply:

- a) If the force identification is the same as that of the receiving entity, the Entity Type field shall be used to display the issuing entity (so that its friendly force appearance will be shown if guises are in use).
- b) If the force identification is different from that of the receiving entity, the Alternate Entity Type field shall be used to display the issuing entity (so that its opposing force appearance will be shown if guises are in use).

4.5.2.1.5.3 Example of Guise function

The Guise function might be employed in the following manner:

In a simulation exercise, the friendly force appearance is an M1 tank, the opposing force appearance is a T-72. All participants would use an entity type of M1 and an alternate entity type of T-72 in the Entity State PDUs they issue.

During the exercise, members of Team 1 receiving Entity State PDUs from their own team will depict them using the friendly force appearance M1. Entity State PDUs received from members of Team 2 will be depicted by members of Team 1 as T-72s by using the opposing force appearance.

Likewise, members of Team 2 receiving Entity State PDUs from their own team will depict them using the friendly force appearance M1. Entity State PDUs received from members of Team 1 will be depicted by members of Team 2 as T-72s.

4.5.2.2 Collision PDU

The Collision PDU shall be used to communicate information about a collision between two simulated entities or between a simulated entity and another object in the simulated world (e.g., a cultural feature such as a bridge or building).

4.5.2.2.1 Collision event

A collision shall be defined as an event that occurs when all of the following conditions are true:

- a) Boundary volumes of two simulated entities intersect (one of which may be a terrain object).
- b) Distance between the origins of the two simulated entities is decreasing.
- c) At least one of the entities is moving at a speed greater than COLLISION_THRSH_DFLT.

4.5.2.2.2 Information contained in the Collision PDU

The Collision PDU shall contain the following information:

- a) Identification of the entity that issued the PDU
- b) Identification of the entity with which the issuing entity collided (If this identification number is unknown, the Identification field shall contain ENTITY_ID_UNKNOWN.)
- c) Event identification of the specific event marked by the collision of the entities
- d) Information for damage determination. This information, when available, shall be used by each entity to determine the extent of damage received during the collision, including
 - 1) Velocity vector of the issuing entity
 - 2) Mass of the issuing entity
 - 3) Location of impact in entity coordinates of the entity with which the issuing entity collided

Any of these three fields may be set to zero if data required for that field cannot be determined by the issuing entity.

- e) Information identifying whether the collision should be modeled as an elastic or inelastic type collision

4.5.2.2.3 Issuance of the Collision PDU

The Collision PDU shall be issued by an entity when a collision is detected between the issuing entity and an object or some other entity taking part in the simulation exercise. If the collision involves two entities, both entities shall issue the Collision PDU even if only one of them detected the collision. An entity that receives a Collision PDU indicating another entity has collided with it without first detecting such a collision shall issue a Collision PDU naming the entity that issued the first Collision PDU. Steps should be taken to ensure that if and when it subsequently detects the same collision event it does not generate a Collision PDU to report it. When a simulation application receives a Collision PDU naming an entity it simulates as the other party involved in a collision after reporting the same collision event, it shall not send another Collision PDU in response. A simulation application shall always issue one, and only one, Collision PDU per collision event it detects or is informed about in which an entity it simulates is a participant, even if that application does not perform collision detection tests.

The Collision PDU shall be issued by using a best effort multicast communication service.

4.5.2.2.4 Receipt of the Collision PDU

Upon receipt of the Collision PDU, the data contained therein shall be used to record the event and to determine the extent of the damage sustained in the collision.

4.5.3 Warfare

Information associated with the representation of weapons effects in a DIS exercise shall be communicated through the use of two PDUs—the Fire PDU and the Detonation PDU (see also 5.3.4).

4.5.3.1 Representation of weapons fire in DIS

Representation of weapons fire in a DIS exercise shall consist of the following sequence of events:

- a) *An entity fires a weapon.* The firing of a weapon shall be communicated through the use of a Fire PDU.
- b) *Munition is launched.* The munition launched shall be modeled by the simulation application controlling the munition. If tracking data is required for the munition, it shall be assigned a unique entity identifier by the firing entity's simulation application. In addition to issuing the Fire PDU, the simulation application modeling the munition's behavior shall issue Entity State PDUs for the munition according to the procedures for the use of the Entity State PDU (see 4.5.2). The munition, therefore, is represented as an entity. If tracking data is not required for the munition, the Munition ID field in the Fire PDU shall be MUNITION_NOT_TRACKED, and no Entity State PDUs shall be issued for this munition. Tracking data should be provided for a munition if representing its travel between firing and impact would effect the outcome of the simulation. This is generally the case if simulation entities can detect and react to the munition during the munition's travel.
- c) *Impact or detonation is communicated.* The impact or detonation of a munition shall be communicated through the use of a Detonation PDU. If the munition is not represented as an entity, detonation represents the end of its path. When a munition is represented as an entity, the termination of the existence of the munition entity is determined by the State bit in the Appearance field set to Deactivated in the munition entity's Entity State PDU.

4.5.3.2 Fire PDU

The Fire PDU shall be used to communicate information associated with the firing of a weapon.

4.5.3.2.1 Information contained in the Fire PDU

The Fire PDU shall contain the following information effective at the instant the munition enters the simulated world:

- a) Identification of the entity issuing the PDU
- b) Identification of the intended target entity if known to the simulation application, otherwise, TARGET_ID_UNKNOWN
- c) Identification of tracked munitions (munitions not tracked shall have a munition identification value of MUNITION_NOT_TRACKED)
- d) Event identification of the specific event marked by the firing of an entity's weapon
- e) Identification of the fire mission if known, otherwise NO_FIRE_MISSION
- f) Information required for representation of the path and impact of the munition, including
 - 1) Location from which the munition was launched or fired
 - 2) Type of munition fired

- 3) Warhead of the munition (if applicable, otherwise, it has a value of zero)
- 4) Fuse employed by the munition (if applicable, otherwise, it has a value of zero)
- 5) Quantity and rate at which munition was fired
- 6) Initial velocity of the munition when visible effects of the launch first become apparent
- 7) Range (three dimension, straight-line distance) that the firing entity's fire control system has assumed for computing the fire control solution

The PDU shall be timestamped with the time the munition clears the firing weapon.

4.5.3.2.2 Issuance of the Fire PDU

The Fire PDU shall be issued by an entity at the moment it fires a weapon. If the number of rounds to be fired is not known when the first round clears the firing entity, then the number of committed rounds shall be sent. Subsequent Fire PDUs shall report the number of rounds in subsequent bursts, or in half second bursts in the case of continuous fire.

The Fire PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

4.5.3.2.3 Single rounds and bursts of fire

If the firing of the weapon represents a single round, the Quantity field of the Fire PDU shall contain a value of one, and the Rate field shall contain a value of zero. If the firing of a weapon or a group of weapons oriented in the same direction represents multiple rounds, the fields shall contain the quantity of munition fired and the rate at which it was fired, respectively.

4.5.3.2.4 Receipt of the Fire PDU

Upon receipt of a Fire PDU, a simulation application shall use the information therein to represent any necessary visual and aural effects produced by the firing of the weapon, whether it be a muzzle flash, noise, or smoke.

4.5.3.3 Detonation PDU

The Detonation PDU shall be used to communicate information associated with the impact or detonation of a munition.

4.5.3.3.1 Information contained in the Detonation PDU

The Detonation PDU shall contain the following information:

- a) Identification of the entity issuing the PDU.
- b) Identification of the target entity if an entity is impacted, otherwise, NO_ENTITY_IMPACTED.
- c) Identification of the munition being detonated. This identification shall be the same as the Munition ID in the Fire PDU for detonations preceded by a fire event, or the Entity ID in an Entity State PDU for detonations not preceded by a Fire PDU.
- d) Identification of the fire event responsible for the detonation. This shall be the same as the Event ID from the corresponding Fire PDU. If the detonation is not preceded by a corresponding fire event, then the Event Number field of the Event Identifier record shall be zero (e.g., land mine's detonation).
- e) Information required for representation of the impact or detonation of the munition, including
 - 1) Location with respect to the world
 - 2) Type of munition fired

- 3) Warhead of the munition (if applicable, otherwise, it has a value of zero)
- 4) Fuse employed by the munition (if applicable, otherwise, it has a value of zero)
- 5) Quantity and rate at which the munition was fired
- 6) Velocity just before detonation/impact
- 7) Location of detonation with respect to the target entity
- 8) Detonation result
- 9) Articulation parameters for any articulated parts of the target entity affected by the detonation

4.5.3.3.2 Issuance of the Detonation PDU

The Detonation PDU shall be issued by a simulation application at the moment that a munition being modeled by that simulation application impacts the terrain, a terrain feature, or another entity, or at the moment it detonates. If the munition neither impacts nor detonates, the controlling simulation application shall issue a Detonation PDU with a detonation result of *none* when the controlling simulation application has ceased to model the munition.

The Detonation PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

4.5.3.3.3 Interpretation of detonation result and inclusion of entity identifier

If the impact or detonation is known to have affected only a specific entity that was physically contacted by the munition, a detonation result of *entity impact* shall be used. If the impact or detonation is known to have affected only a specific entity, but that entity was not physically contacted by the munition prior to detonation, a detonation result of *entity proximate detonation* shall be used. In either case, the entity identification of the specific entity affected shall be included in the Target Entity ID field of the Detonation PDU. The location of the impact or detonation shall be specified in both world coordinates (in the Location in World Coordinates field) and in the entity coordinates of the affected entity (in the Location in Entity's Coordinates field).

If the impact or detonation is known to have affected only the terrain and the terrain was physically contacted by the munition, a detonation result of *surface impact* shall be used. If the impact or detonation is known to have affected only the terrain but the terrain was not physically contacted by the munition prior to detonation, a detonation result of *surface proximate detonation* shall be used. In either case, the location of the impact or detonation shall be communicated in world coordinates (in the Location in World Coordinates field). The Location in Entity's Coordinates field shall hold the value [0.0, 0.0, 0.0] and the Target Entity ID field shall hold the value NO_SPECIFIC_ENTITY.

If neither a specific entity nor the terrain is affected by the munition, a detonation result of *detonation* shall be used if the munition detonates, and a detonation result of *none* shall be used if it does not. In either case the Location in World Coordinates field shall hold the terminal location of the munition. The Location in Entity's Coordinates field shall contain the value [0.0, 0.0, 0.0], and the Target Entity ID field shall contain the value NO_SPECIFIC_ENTITY.

4.5.3.3.4 Inclusion of articulated part parameters

When the firing entity determines that its round has impacted an articulated part on the target entity, the firing entity shall include the articulation parameters (see annex A) of the affected articulated part (at the time of impact) in the Detonation PDU. If the articulated part is not directly connected to the base model, all intervening articulations shall be included in the Detonation PDU. The articulation parameters shall be represented in the same form as in the Entity State PDU.

4.5.3.3.5 Termination of the existence of munition entities

The termination of the existence of a munition entity shall be indicated by the issuance of an Entity State PDU with the State bit in the Appearance field set to Deactivated.

4.5.3.3.6 Receipt of the Detonation PDU

Upon the receipt of a Detonation PDU, a simulation application shall use the information therein to represent the visual and aural effects that may be produced by the detonation or impact of the munition. The receiving simulation application shall also use the information to determine damage that may have been received as a result of the detonation. The simulation application shall issue an Entity State PDU to communicate any appearance changes resulting from the detonation.

4.5.4 Logistics

Information associated with the representation of logistics support in a DIS exercise shall be communicated through the use of several PDUs—Service Request PDU, Resupply Offer PDU, Resupply Received PDU, Resupply Cancel PDU, Repair Complete PDU, and Repair Response PDU (see 5.3.5).

The procedures associated with logistics support require the definition of several timers. The value of these timers shall be established at exercise start and may be changed during the exercise by means of Simulation Management PDUs. If no values are established, the default values for these timers shall be as follows:

- | | | |
|----|----------------------------|-------------------|
| a) | Resupply Receive Timer 1: | RESUP_REC_T1_DFLT |
| b) | Resupply Receive Timer 2: | RESUP_REC_T2_DFLT |
| c) | Resupply Supplier Timer 1: | RESUP_SUP_T1_DFLT |
| d) | Repair Receive Timer 1: | REPAR_REC_T1_DFLT |
| e) | Repair Supplier Timer 1: | REPAR_SUP_T1_DFLT |
| f) | Repair Supplier Timer 2: | REPAR_SUP_T2_DFLT |

4.5.4.1 Procedure for logistics support

Logistics support in DIS shall be accomplished through a series of request and response messages between two entities. Two types of service have been defined for DIS—resupply and repair.

4.5.4.2 State information for resupply service

The different states and transitions for resupply service are described in 4.5.4.2.1 and 4.5.4.2.2. An example of the resupply function is given in 4.5.4.8.

4.5.4.2.1 Receiving entity

The receiving entity may be in one of three states:

- Ready state.* A receiving entity is in the Ready state when it is not in either the Requesting state or the Receiving state.
- Requesting state.* A receiving entity is in the Requesting state when it has requested supplies and has not received a reply to its request.
- Receiving state.* A receiving entity is in the Receiving state when it has been offered supplies and is in the process of receiving them.

The behavior of the receiving entity during resupply service is shown in figure 4. The associated state transitions for the receiving entity are as defined in table 1.

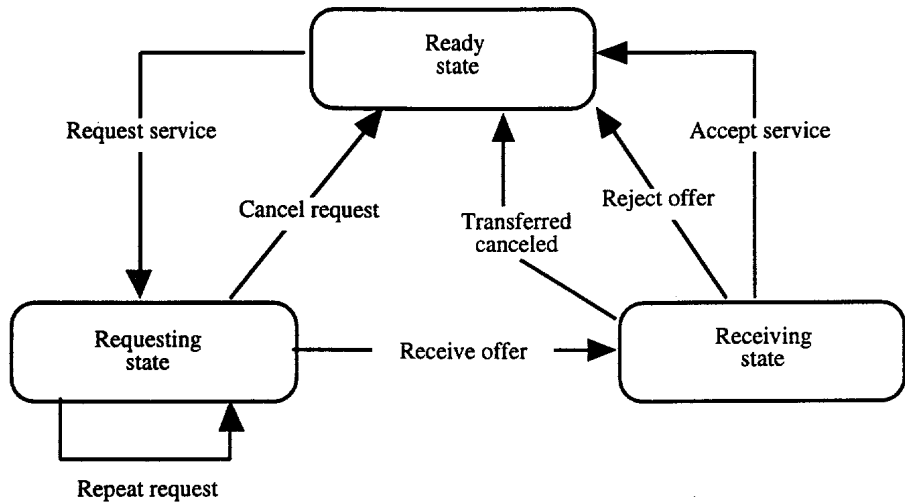


Figure 4—Receiving entity behavior during resupply

Table 1—Resupply state transitions for a receiving entity

Transition	Condition and actions
Request service	When conditions for resupply are met, the entity shall issue a Service Request PDU (see 4.5.4.4). The entity shall proceed from the Ready state to the Requesting state and Resupply Receiver Timer 1 shall be set. ^a
Cancel request	When conditions for resupply are no longer met or when a Resupply Cancel PDU (see 4.5.4.7) is received from a supplying entity, Resupply Receiver Timer 1 shall be canceled and the entity shall proceed from the Requesting state to the Ready state.
Repeat request	When Resupply Receiver Timer 1 expires, the Service Request PDU shall be reissued and the timer shall be reset.
Receive offer	When a Resupply Offer PDU (see 4.5.4.5) is received, Resupply Receiver Timer 1 shall be canceled and Resupply Receiver Timer 2 shall be set to the period of time required for receiving some portion of the offered supplies. ^b The entity shall proceed from the Requesting state to the Receiving state.
Reject offer	When conditions for resupply are no longer met, a Resupply Cancel PDU shall be issued, Resupply Receiver Timer 2 shall be canceled, and the entity shall proceed from the Receiving state to the Ready state.
Accept service	When Resupply Receiver Timer 2 expires, the count of supplies on board shall be incremented and a Resupply Received PDU (see 4.5.4.6) shall be issued. The entity shall then proceed from the Receiving state to the Ready state.
Transfer canceled	When a Resupply Cancel PDU (see 4.5.4.7) is received, the ongoing transfer shall be canceled, Resupply Receiver Timer 2 shall be canceled, and the entity shall proceed from the Receiving state to the Ready state. No supplies (from the canceled transfer) shall be transferred.

^aThe fact that an entity is able to provide resupply or repair service is indicated in the Capabilities field of its Entity State PDU.

^bResupply Receive Timer 2 shall be set to a value less than Resupply Supplier Timer 1. The difference between these two timers shall be sufficient to account for network and processing delays. A minimum difference of 5 s is suggested.

4.5.4.2.2 Supplying entity

The supplying entity may be in one of two states:

- Ready state.* A supplying entity is in the Ready state when it is able to receive a request for supplies and is able to offer supplies to a receiving entity.
- Offering state.* A supplying entity is in the Offering state when it has made an offer of supplies and is waiting for the receiving entity to indicate the quantity of supplies it has taken.

The behavior of the supplying entity during resupply service is shown in figure 5. The state transitions for the supplying entity are as defined in table 2.

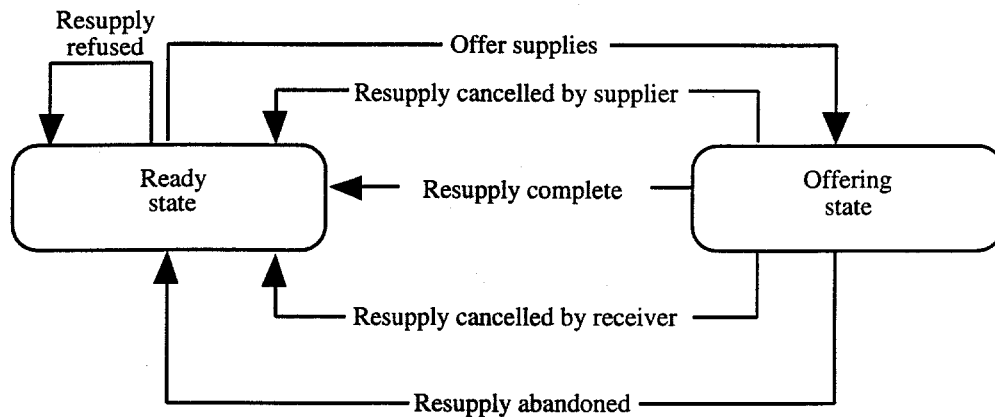


Figure 5—Supply entity behavior during resupply

Table 2—Resupply state transitions for a supplying entity

Transition	Condition and actions
Offer supplies	When a Service Request PDU (see 4.5.4.4) is received and conditions for resupply are met, a Resupply Offer PDU (see 4.5.4.5) shall be issued, Resupply Supplier Timer 1 shall be set, and the entity shall proceed from the Ready state to the Offering state.
Resupply refuse	When a Service Request PDU (see 4.5.4.4) is received and the requested supplies are unavailable, a Resupply Cancel PDU (see 4.5.4.7) shall be issued. The entity shall remain in the Ready state.
Resupply complete	When a Resupply Received PDU (see 4.5.4.6) is received, Resupply Supplier Timer 1 shall be canceled. The count of supplies on board shall be decremented and the entity shall proceed from the Offering state to the Ready state.
Resupply cancelled by receiver	When a Resupply Cancel PDU (see 4.5.4.7) is received, Resupply Supplier Timer 1 shall be canceled. The count of supplies on board shall not change and the entity shall proceed from the Offering state to the Ready state.
Resupply cancelled by supplier	When conditions for resupply are no longer met, a Resupply Cancel PDU (see 4.5.4.7) shall be issued, Resupply Timer 1 shall be cancelled, and the entity shall proceed from the Offering state to the Ready state.
Resupply abandoned	When Resupply Supplier Timer 1 expires, the transfer shall be abandoned, the count of supplies on board shall not change, and the entity shall proceed from the Offering state to the Ready state.

4.5.4.3 State information for repair service

The different states and transitions for repair are described in 4.5.4.3.1 and 4.5.4.3.2. An example of repair service is given in 4.5.4.12.

4.5.4.3.1 Receiving entity

The receiving entity may be in one of two states:

- a) *Ready state.* A receiving entity is in the Ready state when it is able to request repairs from an entity with repair capabilities.
- b) *Requesting/Receiving state.* A receiving entity is in the Requesting/Receiving state when it has requested repairs and has not received a response to its request.

The behavior of the receiving entity during repair service is shown in figure 6. The associated state transitions are described in table 3.

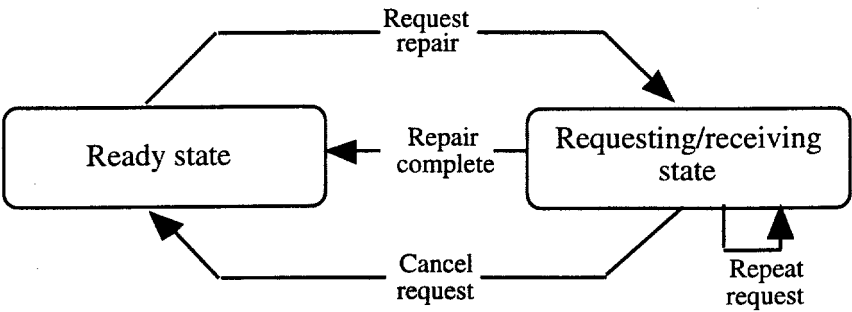


Figure 6—Receiving entity behavior during repair

Table 3—Repair service state transitions for a receiving entity

Transition	Condition and actions
Request repair	When conditions for repair service are met, the entity shall issue a Service Request PDU (see 4.5.4.4), Repair Receiver Timer 1 shall be set, and the entity shall proceed from the Ready state to the Requesting/Receiving state.
Repair complete	When a Repair Complete PDU (see 4.5.4.9) is received, the entity shall issue a Repair Response PDU (see 4.5.4.10), Repair Receiver Timer 1 shall be canceled, and the entity shall proceed from the Requesting/Receiving state to the Ready state.
Cancel request	When conditions for repair service are no longer met, the receiving entity shall cease to issue Service Request PDUs, Repair Receiver Timer 1 shall be canceled, and the entity shall proceed from the Requesting/Receiving state to the Ready state.
Repeat request	When Repair Receiver Timer 1 expires, the Service Request PDU shall be reissued and the time reset. The entity shall remain in the Requesting/Receiving state.

4.5.4.3.2 Repairing entity

The repairing entity may be in one of three states:

- Ready state.* A repairing entity is in the Ready state when it is able to offer repairs to a receiving entity.
- Offering state.* A repairing entity is in the Offering state when it has received a request for repairs and is responding to the request.
- Repair Completed state.* A repairing entity is in the Repair Completed state when it has completed a repair and it is waiting for a response.

The behavior of the repairing entity during repair service is shown in figure 7. Repair service state transitions for a repairing entity are described in table 4.

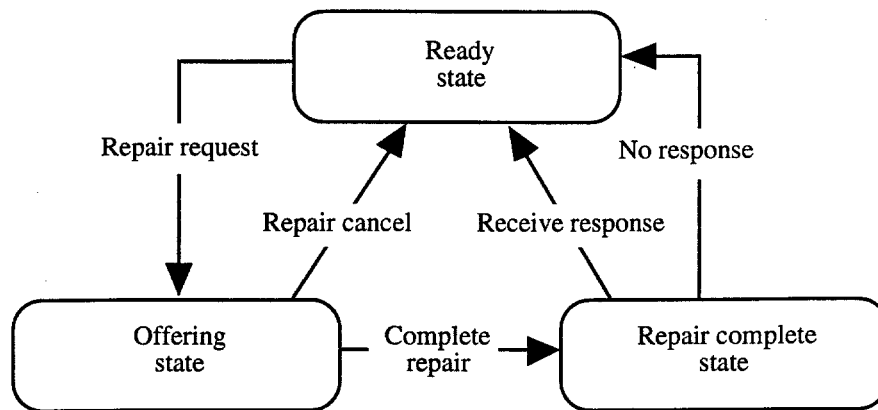


Figure 7—Repairing entity behavior during repair

Table 4—Repair service state transitions for a repairing entity

Transition	Condition and actions
Repair request	When the repairing entity has received a Service Request PDU (see 4.5.4.4) and has identified itself as the intended repairing entity, Repair Supplier Timer 1 shall be set. Every time a Service Request PDU is received from the entity requesting the repair service, Repair Supplier Timer 1 shall be reset. The repairing entity shall then proceed from the Ready state to the Offering state.
Complete repair	When the repair is complete, the entity shall issue a Repair Complete PDU (see 4.5.4.9), shall cancel Repair Supplier Timer 1, shall set Repair Supplier Timer 2, and shall proceed from the Offering state to the Repair Completed state.
Receive response	When a Repair Response PDU is received from the receiving entity, the entity shall cancel the Repair Supplier Timer 2 and shall proceed from the Repair Completed state to the Ready state.
No response	When Repair Supplier Timer 2 has expired, the entity shall proceed from the Repair Completed state to the Ready state.
Repair canceled	When Repair Supplier Timer 1 has expired, the entity shall proceed from the Offering state to the Ready state.

4.5.4.4 Service Request PDU

The Service Request PDU shall be used to communicate information associated with one entity requesting a service from another.

4.5.4.4.1 Information contained in the Service Request PDU

The Service Request PDU shall contain the following information:

- a) Identification of the entity issuing the PDU
- b) Identification of the entity that is able to provide the service required by the requesting entity
- c) Type of service being requested. Services defined are:
 - 1) Resupply
 - 2) Repair
- d) Number and types of supplies if the service required is resupply

4.5.4.4.2 Issuance of the Service Request PDU

The Service Request PDU shall be issued by an entity requesting logistics support when appropriate conditions exist. Appropriate conditions include internal conditions (such as certain crew actions in the simulator), as well as external conditions (such as conditions existing in the simulated world, entities not destroyed, or being within a certain distance of the entity requesting logistics support).

The Service Request PDU shall be issued using a best effort multicast communication service.

4.5.4.4.3 Receipt of the Service Request PDU

Upon receipt of a Service Request PDU, the entity that receives the PDU shall respond in one of the following ways:

- a) If the service requested is resupply and the entity that receives the PDU is able to provide the needed supplies, the supplying entity shall issue a Resupply Offer PDU (see 4.5.4.5).
- b) If the service requested is repair, the repairing entity shall simulate the needed repairs in the following manner: The repair process is allowed to proceed as long as the repairing entity continues to receive Service Request PDUs. If Service Request PDUs cease to be received and are not seen for the period of Repair Supplier Timer 1, the repairing entity shall assume that conditions for repair no longer exist and therefore shall abort the process.
- c) If the service requested is resupply and the entity receiving the PDU is unable to provide the supplies requested, then the supplying entity shall issue a Resupply Cancel PDU (see 4.5.4.7).

4.5.4.5 Resupply Offer PDU

A Resupply Offer PDU shall be used to communicate the offer of supplies by a supplying entity to a receiving entity.

4.5.4.5.1 Information contained in the Resupply Offer PDU

The Resupply Offer PDU shall contain the following information:

- a) Identification of the entity requesting resupply
- b) Identification of the supplying entity that issued the PDU
- c) Number of types of supplies that the supplying entity is able to provide
- d) Supply types available and the amount of each

4.5.4.5.2 Issuance of the Resupply Offer PDU

The Resupply Offer PDU shall be issued by an identified supplying entity that has received a Service Request PDU (see 4.5.4.4) requesting resupply service.

The Resupply Offer PDU shall be issued using a best effort multicast communication service.

4.5.4.5.3 Receipt of the Resupply Offer PDU

Upon receipt of a Resupply Offer PDU, the receiving entity shall proceed from the Requesting state to the Receiving state. When receipt of the supplies is complete, the receiving entity shall respond by issuing a Resupply Received PDU.

4.5.4.6 Resupply Received PDU

A Resupply Received PDU shall be used to acknowledge the receipt of supplies by the receiving entity.

4.5.4.6.1 Information contained in the Resupply Received PDU

The Resupply Received PDU shall contain the following information:

- a) Identification of the entity requesting resupply
- b) Identification of the supplying entity
- c) Number of types of supplies that the receiving entity took
- d) Supply types available and the amount of each taken by the receiving entity

4.5.4.6.2 Issuance of the Resupply Received PDU

The Resupply Received PDU shall be issued by an identified receiving entity to indicate the supplies actually transferred from the supplying entity to the receiving entity.

The Resupply Received PDU shall be issued using a best effort multicast communication service.

4.5.4.6.3 Receipt of the Resupply Received PDU

Upon receipt of a Resupply Received PDU, the supplying entity shall decrement the number of supplies on board and shall proceed to the Ready state.

4.5.4.7 Resupply Cancel PDU

The Resupply Cancel PDU shall be used to communicate the canceling of a resupply service provided through logistics support.

4.5.4.7.1 Information contained in the Resupply Cancel PDU

The Resupply Cancel PDU shall contain the following information:

- a) Identification of the entity receiving supplies
- b) Identification of the entity providing supplies

4.5.4.7.2 Issuance of the Resupply Cancel PDU

The Resupply Cancel PDU shall be issued by either the receiving entity or supplying entity at any time during resupply to cancel the resupply service.

The Resupply Cancel PDU shall be issued using a best effort multicast communication service.

4.5.4.7.3 Receipt of the Resupply Cancel PDU

The entity receiving the Resupply Cancel PDU shall cancel its timers and return to the Ready state. No supplies (for the canceled transaction) are transferred.

4.5.4.8 Example of resupply service

If the service requested by the receiving entity is resupply, a scenario such as the following may take place:

A resupply interaction begins when a simulation application determines that one of its entities is in a state such that it should be resupplied. For example, a ground vehicle might come to a stop and open its refueling port. When this occurs, the simulation application issues a Service Request PDU to a potential resupplying entity and sets Resupply Receiver Timer 1. Potential resuppliers may be identified by their proximity, state, and the content of the Capabilities field (as determined from their Entity State PDUs). This event is shown as the request service transition in figure 4.

A simulation entity that receives a Service Request PDU and is identified as the specified supplying entity responds by offering some portion of whatever supplies are currently loaded on board. This condition is shown in figure 5 as a transition from the Ready state to the Offering state. Meanwhile, the receiving entity reissues its Service Request PDU every time Resupply Receiver Timer 1 expires until such an offer is forthcoming. The offer takes the form of a Resupply Offer PDU issued by the supplying entity. The supplies offered should be a subset of those possessed by the supplying entity, and a subset of those requested by the receiving entity.

Upon receiving the offer of supplies, the receiving entity changes from the Requesting state to the Receiving state. The receiving entity then has until the Resupply Receiver Timer 2 expires to acknowledge the receipt of those supplies by returning to the supplying entity a Resupply Received PDU listing the exact supplies taken. The receiving entity need not accept all of the supplies offered, but instead can indicate in its receipt how many it did accept. After delaying up to the time when Resupply Receiver Timer 2 expires, the receiving entity issues its Resupply Received PDU and returns to the Ready state. When the supplying entity receives the Resupply Received PDU, it also returns to the Ready state, and the procedure is complete.

The time required to return the Resupply Received PDU, and the quantity of supplies reported by that PDU as taken determine the rate at which the supplying entity and the receiving entity are able to transfer munitions. For example, an M1 tank obtaining 105 mm shells from an ammunition supply truck might acknowledge receipt of a single round after 40 s; this results in a simulated rate of resupply for the M1 tank of one round every 40 s.

Throughout the transfer process, both the receiving entity and the supplying entity continue to monitor the conditions necessary for the transfer. If any of these conditions ceases to hold, either entity can abort the transfer by issuing a Resupply Cancel PDU, with the result that no supplies are transferred (for the transfer that was in process). Alternatively, the receiving entity can terminate the transfer early but accept some of the supplies offered by issuing a Resupply Received PDU for the partial load. Finally, if the supplying entity waits in the Offering state for the entire period of Resupply Supplier Timer 1 but receives no Resupply Received PDU (perhaps the receiving entity has withdrawn from the exercise), it should return to the Ready state and assume that no supplies were taken.

4.5.4.9 Repair Complete PDU

The Repair Complete PDU shall be used by the repairing entity to communicate the repair that has been performed for the entity that requested repair service.

4.5.4.9.1 Information contained in the Repair Complete PDU

The Repair Complete PDU shall contain the following information:

- a) Identification of the entity requesting repair service
- b) Identification of the entity providing the repair
- c) Repair performed by the repairing entity. (Possible repairs are included in Section 6 of EBV-DOC.)

4.5.4.9.2 Issuance of the Repair Complete PDU

The Repair Complete PDU shall be issued by a repairing simulation entity upon completion of a repair service requested by the receiving entity in a Service Request PDU (see 4.5.4.4).

The Repair Complete PDU shall be issued by using a best effort multicast communication service.

4.5.4.9.3 Receipt of the Repair Complete PDU

Upon receipt of the Repair Complete PDU, the receiving entity shall issue a Repair Response PDU (see 4.5.4.10), and proceed from the Requesting/Receiving state to the Ready state.

4.5.4.10 Repair Response PDU

A Repair Response PDU shall be used by the receiving entity to acknowledge the receipt of a Repair Complete PDU (see 4.5.4.9).

4.5.4.10.1 Information contained in the Repair Response PDU

The Repair Response PDU shall contain the following information:

- a) Identification of the entity requesting repair service
- b) Identification of the entity providing the repair
- c) Result of the repair

4.5.4.10.2 Issuance of the Repair Response PDU

The Repair Response PDU shall be issued by the entity receiving repair service upon receipt of a Repair Complete PDU from the repairing entity.

The Repair Response PDU shall be issued by using a best effort multicast communication service.

4.5.4.10.3 Receipt of the Repair Response PDU

Upon receipt of the Repair Response PDU, the repairing entity shall note that the receiving entity has received the repair.

4.5.4.11 Cancellation of repair service

If the receiving entity intends to cancel the repair service before the repairs are completed, it shall cease to issue Service Request PDUs and shall return to the Ready state. The supplying entity that does not receive Service Request PDUs for the period of Repair Supplier Timer 1 shall abandon the repair service and shall return to the Ready state. If the supplying entity intends to cancel the repair service, it shall issue a Repair Complete PDU and shall list the repair result as *no repairs performed*. The receiving entity shall respond with the issue of a Repair Response PDU indicating the repair result as *service canceled by the supplying entity*.

4.5.4.12 Example of repair service

If the service requested by the receiving entity is repair, a scenario such as the following may take place:

A simulation application issues a Service Request PDU and sets Repair Receiver Timer 1 when it determines that one of its simulation entities is in need of repairs. The Service Request PDU is issued to an entity capable of performing repairs as determined by its proximity, state, and the content of the field capabilities conveyed in its Entity State PDU. This event is shown as the request repair transition in figure 6.

If the repair process successfully runs to completion, the repairing entity may then accomplish the repair by issuing a Repair Complete PDU to notify the receiving entity of the repair, and returning to the Ready state. The receiving entity's simulation application acknowledges the receipt of the Repair Complete PDU by returning a Repair Response PDU. (This acknowledgment simply indicates that the repair was performed, not that the repair was appropriate or that the disabled vehicle has been made operational because of the repair.)

4.5.5 Simulation management

The content and the procedure for use of Simulation Management PDUs in a DIS exercise are established in 4.5.5.1 through 4.5.5.6.3. The Simulation Management protocol may or may not be required for participation in an exercise, depending upon the requirements of each exercise.

4.5.5.1 Simulation Management PDU header

Certain information shall be included with each Simulation Management PDU. This information includes standard PDU header (see 4.5.1), originating entity (entity issuing the PDU), and receiving entity.

4.5.5.1.1 PDU header

This field shall be required at the beginning of each DIS PDU. It shall contain information about the protocol version, the exercise identification, the type of PDU, the protocol family to which the PDU belongs, a timestamp, and the length of the PDU. PDU types comprising the Simulation Management protocol family are:

- a) Create Entity
- b) Remove Entity
- c) Start/Resume
- d) Stop/Freeze
- e) Acknowledge
- f) Action Request
- g) Action Response
- h) Data query
- i) Set Data
- j) Data
- k) Event Report
- l) Comment

4.5.5.1.2 Entity IDs

The unique identification of each entity in an exercise consists of a Simulation Address record and an Entity Identifier number. See 5.2.14 for detailed descriptions of each of these components.

4.5.5.1.2.1 Originating Entity ID

The identification for the entity responsible for issuing the PDU shall be specified in this field. This field shall be composed of Site Identification number, Application Identification number, and Entity Identifier number.

4.5.5.1.2.2 Receiving Entity ID

The identification of the intended receiving entity shall be specified in this field. This field shall be composed of Site Identification number, Application Identification number, and Entity Identifier number. If the PDU is a Create Entity PDU, the Entity Identifier field of the Receiving Entity ID record shall contain the Entity Identifier number assigned to a new entity if it is known.

If the receiving simulation application is intended to assign the entity identifier to the newly created entity, then the value of the Entity Identifier field of the Receiving Entity ID record in the Create Entity PDU shall be the value RQST_ASSIGN_ID. In any case, the identification of the new entity shall be returned in the originating Entity ID field of the Acknowledge PDU. If the receiving simulation application is unable to comply with the request to assign the entity identifier, the simulation application shall signify this by returning an entity identification with the Entity Identifier field equal to NO_ENTITY in the Originating Entity ID field of the Acknowledge PDU.

4.5.5.2 The simulation management computer

Simulation management may be performed by any computer on the DIS network. The computer may be dedicated to the task of simulation management, it may be hosting a simulation application or an interface to a simulator, or it may be the same computer that is simulating the entity (i.e., running the simulation application). In all cases, the computer that is performing simulation management functions shall be referred to in this standard as the Simulation Manager (SM). An exercise may have one SM, or multiple SMs. One entity may interact with multiple SMs during an exercise, or an entity may act as its own SM and may or may not interact with other SMs during an exercise.

4.5.5.3 Simulation management functions

Simulation management functions can be categorized as entity/exercise management and data management. Many actions that are performed with individual entities can also be performed exercise wide. For example, an entity or an exercise can be initialized, started, or stopped. Many entity/exercise actions are, therefore, explained on the entity level with the understanding that the same actions may also apply on an exercise level as well. Information associated with performing these functions is communicated by a number of different PDUs.

The PDUs required for simulation management are described in 4.5.5.4. See 4.5.5.5 for a discussion of the simulation management functions and how the PDUs shall be used to accomplish them.

4.5.5.4 Simulation management PDUs

The PDUs described in 4.5.5.4.1 through 4.5.5.4.12.3 shall be used to perform simulation management functions.

4.5.5.4.1 Create Entity PDU

The Create Entity PDU shall communicate information about the creation of a new entity for a DIS exercise. This PDU simply establishes the identity of the new entity.

4.5.5.4.1.1 Information contained in the Create Entity PDU

The Create Entity PDU shall contain the same information found in the Simulation Management PDU header and a request identification for the creation action being requested by the SM. The Originating Entity ID shall represent the identification number for the SM that is responsible for creating the new entity. The Receiving Entity ID shall represent the entity identification number of the newly created entity if it is known. Identification numbers shall be unique to each entity.

The SM may request that the application creating the entity assign the entity number. This is accomplished by setting the Entity Identifier field to RQST_ASSIGN_ID. Entity identifications shall be created in accordance with 5.2.14.2.

4.5.5.4.1.2 Issuance of the Create Entity PDU

The Create Entity PDU may be issued by a SM to a simulation application when a new entity is to be created for a DIS exercise. This PDU is not necessary for the creation of munition entities that require tracking information. Entities may enter an exercise without the use of the Simulation Management protocol (see 4.5.5.5.1).

The Create Entity PDU shall be issued using a best effort unicast or multicast communication service.

4.5.5.4.1.3 Receipt of the Create Entity PDU

Upon receipt of the Create Entity PDU, if the receiving simulation application can comply with the Create Entity PDU request, then it shall assign an entity identification number to the new entity as requested by the SM. The receiving simulation application shall respond to the Create Entity PDU by issuing an Acknowledge PDU with the Response Flag field indicating that it can comply with the request, or why it cannot.

4.5.5.4.2 Remove Entity PDU

The Remove Entity PDU shall communicate the removal of an entity from a DIS exercise. This PDU indicates to the receiving entity that it is being removed from the exercise.

NOTE—An entity may affect the removal of itself from an exercise by setting the State bit in the Appearance field equal to Deactivated in the last Entity State PDU that it issues. See 4.5.2.1.3.

4.5.5.4.2.1 Information contained in the Remove Entity PDU

The Remove Entity PDU shall contain the same information found in the Simulation Management PDU header and a request identification for the removal action being requested by the SM. The Originating Entity ID shall identify the SM issuing the Remove Entity PDU. The Receiving Entity ID shall identify the entity that is being removed from the simulation exercise.

4.5.5.4.2.2 Issuance of the Remove Entity PDU

The Remove Entity PDU shall be issued by an SM when a particular entity is to be removed from the simulation.

The Remove Entity PDU shall be issued using a best effort unicast or multicast communication service.

4.5.5.4.2.3 Receipt of the Remove Entity PDU

Upon receipt of the Remove Entity PDU, the receiving simulation application shall immediately cease simulating the entity identified in the Remove Entity PDU and shall remove it from the simulation exercise as quickly as possible. The receiving simulation application shall then acknowledge the receipt of the Remove Entity PDU by issuing an Acknowledge PDU.

The simulation entity that is being removed from the exercise shall broadcast at least one more Entity State PDU following the acknowledgment of the Remove Entity PDU. The last Entity State PDU broadcast by the deactivated entity shall have the State bit of the Appearance field set to Deactivated.

Simulation applications other than the one to which the Remove Entity PDU is addressed should ignore the Remove Entity PDU. Entities interacting with the removed entity will be able to recognize that the removed entity has left the exercise through the reception of the Entity State PDU from the removed entity with the State bit of the Appearance field set to Deactivated and the absence of further Entity State PDUs from the removed entity.

4.5.5.4.3 Start/Resume PDU

The Start/Resume PDU shall be used to communicate to a simulation entity that it is to leave a Stopped/Frozen state and begin participating in a simulation exercise. This PDU indicates to the entity the time it is to join the simulation.

4.5.5.4.3.1 Information contained in the Start/Resume PDU

The Start/Resume PDU shall contain the following information:

- a) Standard simulation management header
- b) Time of day in real-world time (UTC)
- c) Time of day in the simulation exercise
- d) Request identification number for the start/resume being requested

4.5.5.4.3.2 Issuance of the Start/Resume PDU

The Start/Resume PDU shall be issued by an SM to an entity to instruct that entity to proceed from a Stopped/Frozen state to a Simulated state (see 4.5.5.5.3 for more on entity states). The Start/Resume PDU should be sent far enough in advance that any receiving simulation application will have time to comply. The maximum expected time between transmission and reception of PDUs under various conditions is described in IEEE Std 1278.2-1995. An application should consider a Start/Resume PDU that arrives after the designated start time an error condition that may be reported with an Event Report PDU.

The Start/Resume PDU shall be issued using a best effort unicast or multicast communication service.

4.5.5.4.3.3 Receipt of the Start/Resume PDU

Upon receipt of the Start/Resume PDU, the receiving entity shall respond by issuing an Acknowledge PDU. The receiving entity shall leave the Stopped/Frozen state and join the exercise at the real-world time indicated in the PDU. The entity shall simulate the time of day indicated in the Simulation Time field.

4.5.5.4.4 Stop/Freeze PDU

The Stop/Freeze PDU shall be used by the SM to indicate to a simulated entity that it shall leave a Simulating state and enter a Stopped/Frozen state. This PDU shall also indicate the reason that the receiving entity was requested to stop. The Stop/Freeze PDU should be sent far enough in advance that any receiving simulation application will have time to comply. The maximum expected time between transmission and reception of PDUs under various conditions is described in IEEE Std 1278.2-1995. An application that receives a Stop/Freeze PDU after the designated stop time should consider this an error condition that may be reported with an Event Report PDU.

4.5.5.4.4.1 Information contained in the Stop/Freeze PDU

The Stop/Freeze PDU shall contain the following information:

- a) Standard simulation management header
- b) The reason the SM requested that the entity stop simulating
- c) Frozen behavior
- d) Request identification number for the stop/freeze action being requested
- e) Real-world time at which the stop/freeze action is to occur

4.5.5.4.4.2 Internal state of a frozen entity

How the internal state of an entity is affected by a stopped internal simulation clock is outside the scope of this standard, except that if an entity is not updating its simulation based upon received PDUs then no reaction to a Non-Simulation Management PDU is permitted.

4.5.5.4.4.3 Issuance of the Stop/Freeze PDU

The Stop/Freeze PDU shall be issued by an SM to an entity when the SM requests an entity to stop simulating.

The Stop/Freeze PDU shall be issued using a best effort unicast or multicast communication service.

4.5.5.4.4.4 Receipt of the Stop/Freeze PDU

Upon receipt of the Stop/Freeze PDU, the receiving simulation application shall respond by issuing an Acknowledge PDU. The receiving entity shall leave the Simulating state at the real-world time indicated in the PDU. The stop/freeze condition may be temporary or indefinite based on the reason for stop/freeze.

4.5.5.4.5 Acknowledge PDU

The Acknowledge PDU shall be used to acknowledge the receipt of a Create Entity PDU, a Remove Entity PDU, a Start/Resume PDU, or a Stop/Freeze PDU. This PDU verifies to the SM the receipt of the issued PDU. The Acknowledge PDU shall be sent as soon as possible after the receipt of the above PDUs.

4.5.5.4.5.1 Information contained in the Acknowledge PDU

The Acknowledge PDU shall contain the following information:

- a) Standard simulation management header. The Originating Entity ID refers to the entity issuing the Acknowledge PDU. The Receiving Entity ID shall refer to the SM to whom the entity is responding.
- b) A flag to indicate the type of PDU message to which the acknowledgment pertains.
- c) Request identification number for the acknowledgment being sent to the SM. The request identification number shall match the request identification number from the PDU that is being acknowledged.
- d) A response flag to indicate the level to which the entity was able to comply with the action being acknowledged.

4.5.5.4.5.2 Issuance of the Acknowledge PDU

The Acknowledge PDU shall be issued by a simulation application in response to a Create Entity PDU, a Remove Entity PDU, a Start/Resume PDU, or a Stop/Freeze PDU.

The Acknowledge PDU shall be issued using a best effort unicast or multicast communication service.