

IEEE Std 1278.1a-1998
(Supplement to IEEE Std 1278.1-1995)

IEEE Std 1278.1a-1998

IEEE Standard for Distributed Interactive Simulation— Application Protocols

IEEE Computer Society

Sponsored by the
Distributed Interactive Simulation Committee

19 August 1998

SH94604

IEEE Standard for Distributed Interactive Simulation— Application Protocols

Sponsor

**Distributed Interactive Simulation Committee
of the
IEEE Computer Society**

Approved 19 March 1998

IEEE-SA 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, Radio Communications, Entity Management, Minefield, Synthetic Environment, Simulation Management with Reliability, Live Entity Information/Interaction, and Non-Real Time.

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

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

Copyright © 1998 by the Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 1998. Printed in the United States of America.

ISBN 0-7381-0174-5

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 IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) 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 societies and Standards Coordinating 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-SA 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 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; (978) 750-8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

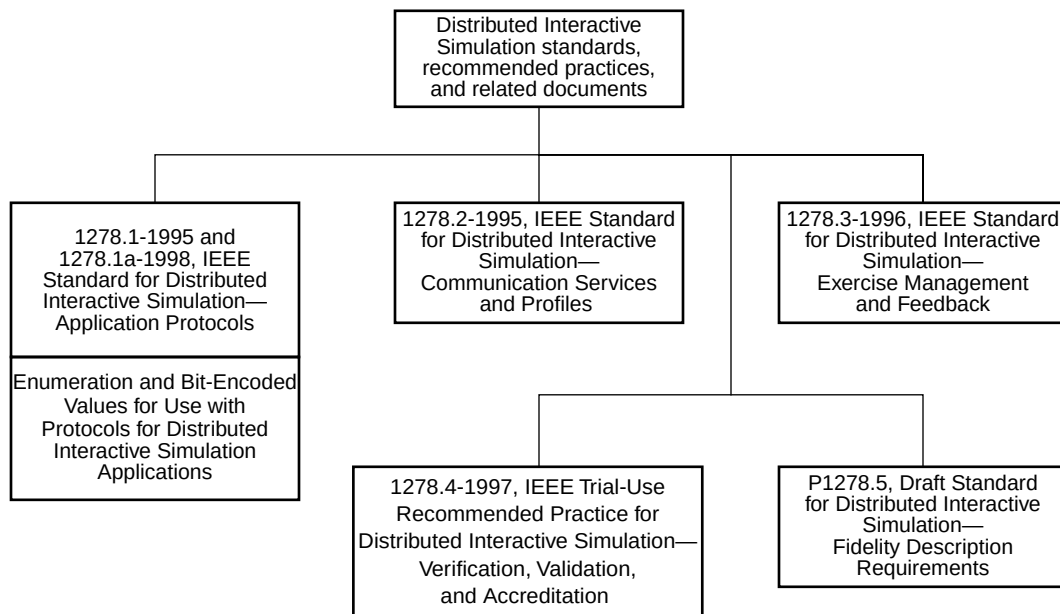
Introduction

(This introduction is not part of IEEE Std 1278.1a-1998, 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 Simulation 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 workshops resulted in several IEEE standards and recommended practices.

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



Documentation relationships

The interoperability components addressed by these standards and recommended practices are:

- a) Application protocols
- b) Communication services and profiles
- c) Exercise management and feedback
- d) Validation and verification
- e) Fidelity description requirements

IEEE Std 1278.1a-1998 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, types of entity interactions that take place in a DIS exercise, data for management and control of a DIS exercise, simulated environment states, aggregation of entities, and the transfer of control of entities. IEEE Std 1278.1a-1998 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 and IEEE Std 1278.1a-1998. This document is entitled *Enumeration and Bit-Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications* and is available from the Defense Modelling Simulation and Technology Information Analysis Center at the Institute for Simulation and Training of the University of Central Florida, Orlando Florida.

IEEE Std 1278.2-1995 defines the communication services required to support the message exchange described in IEEE Std 1278.1-1995 and IEEE Std 1278.1a-1998. In addition, IEEE Std 1278.2-1995 provides several communication profiles that meet the specified communications requirements.

Together IEEE Std 1278.1-1995, IEEE Std 1278.1a-1998, and IEEE Std 1278.2-1995 provide the necessary information exchange for the communications element of DIS.

IEEE Std 1278.3-1996 provides guidelines for establishing a DIS exercise, managing the exercise, and providing proper feedback. This recommended practice is used in conjunction with IEEE Std 1278.1-1995, IEEE Std 1278.1a-1998, and IEEE Std 1278.2-1995.

IEEE Std 1278.4-1997 assists in the planning and conduct of validation and verification processes associated with DIS exercises. This recommended practice presents data flow and connectivity for all proposed validation and verification process model steps and provides rationale and justification for each step.

IEEE Std P1278.5 provides a taxonomy of fidelity characteristics with a consistent and uniform language to describe fidelity at six levels: resource, fidelity domain, capability, implementation, characteristic, and descriptor.

At the time this standard was completed, the Protocols Working Group had the following membership, and was assisted by members of other working groups as listed below.

J. Joseph Brann, Chair

Dale Anglin	Peter Hoare	Dean P. Ringressy
Eric Anschuetz	Carl Hobson	William Rininger
Bradley Babin	Pamela M. Jacobs	Braulio Rivera
Michael Bachman	Patrice Jouanneau	Peter Rizik
Joseph Batman	Daniel Katz	David W. Roberts
Steven Bishop	Warren Katz	John Robins
Benjamin Blood	James Keenan	Michael Robkin
Keith Briggs	Christopher Kennedy	Jerome Sanders
Brett Butler	James Kester	Randy Saunders
Richard Carpenter	Moshe Kirsh	Christian Sautereau
Eric R. Christensen	Samuel N. Knight	Richard Schaffer
Ann Davies	Robert Knox	Eric O. Schmidt
William De Jong	James Kverderis	David Scott
Claude De Medeuil	Joseph Lacetera	Timothy Shaw
Michael Donnelly	Jay Levenson	Steven Sheasby
Kenneth Doris	Todd Liddiard	Steven Slosser
Michael Drinkwater	Rodney Long	Larry Smith
Ursela Eberley	Gerald Lucha	Neil Smith
Jean-Pierre Faye	Paul Maassel	Steven A. Specht
Grayden Figart	Ted Marz	Michael Stalford
Mark Flanagan	Mark McAuliffe	Kenneth Starr
Jerry Forbes	Edward M. McCall	Martin R. Stytz
Darrell Fowler	David McKeeby	William Sudnikovich
Zach Furness	Timothy McKelvy	Kenneth Sullivan
Patrick Getchell	Thomas McLean	Steven Swaine
Sam Giambarberee	John Mills	Darrin Taylor
Stephen Gildner	Steven L. Monson	David Taylor
Kent Glovanovitz	Monte Montierth	James Totten
Suresh Goel	Thomas Mullins	Chris Waguespack
Hector Gonzalez	Franklin Neal	Douglas Wahrenberger
Robert Goodwin	Eric Oelschlager	David Welsh
William Gordon	William Parrish	Roger Werner
Paul Gustavson	Joseph Pasirstein	Jeffery Wicks
Scott Hamilton	David Peden	Debra Wilbert
Susan Harkrider	Barbara J. Pemberton	Edward Williams
Edward Harvey	Jay Pochop	Michael Winburn
Edward Hayes	Lawrence Rieger	Kevin Wire
Gary Hessler		Charles Woodhouse

Emissions

Sam Knight, Subgroup Chair

Janota Claus	James Hammond	Jerome Sanders
Kevin Correll	Daniel Katz	Richard Schaffer
Rudolph Croteau	Robert Kelly	Graham Shanks
David DeFanti	James Kester	Claudia Slaton
Claude De Medeuil	Vincent Kijowski	Steven Slosser
Jerry Denver	Timothy Magnani	Larry Smith
Kenneth Doris	Cathy Matthews	Mike Sutton
Jean-Pierre Faye	Mark McAuliffe	Jeff Swauger
Eric Feathers	David McKeeby	Stephen Swenson
Grayden Figart	Richard Oberle	Darrin Taylor
Jerry Forbes	Brian Petrasko	David Taylor
Richard Gagan	David Quam	Lorraine Tuliano
Ray Giroux	Douglas Reif	Roger Werner
Kent Glovanovitz	Robin Rouleau	Douglas Wood
David Greschke	Terri Rowan	William Zimmerman

Communication Representation and Data Link Representation

Daniel Katz, *Subgroup Co-Chair*

Joseph Lacetera, *Subgroup Co-Chair*

Robert Butterfield
Eric Christensen
Ken Ciszweski
Larry Dworkin
Mark Flanagan
Gerald Forbes

Lynnore Garofolo
Patrick Getchell
Kent Glovanovitz
Frank Hill
Steven L. Monson

Eric Oelschlager
Rob Rand
Richard Schaffer
Tim Schoeny
Larry Smith
Charlie Woodhouse

Simulation Management

Michael Robkin, *Subgroup Chair*

Eric Anschuetz
Charolotte Blair
John Bray
Richard Carpenter
Ken Castner
Dean Daugherty
Ann Davies
John DiCola
Michael Donnelly
Jean Pierre Floch
Suresh Goel
William Gordon
Suraiya Haque
Lewis Harmon

Michael Healy
Gary Hessler
Pamela M. Jacobs
Darrell Jones
Peter Kassal
Barbara Kenney
Joseph King
Robert Knox
Joanne Matone
Thomas McLean
Linda Myers
Franklin Neal
Kimberly Neff
Joseph Pasirstein
Steven Ruch

Nick Rule
Ramesh Sadasivam
Christian Sautereau
Judy Schandua
Isiah Shepherd
W. Garth Smith
Marchelle Stahl
Martin R. Stytz
Michael Tackaberry
David Taylor
James Totten
John Vanderburgh
Gregory Weidman
Debra Wilbert

Synthetic Environment Working Group

Ronald Moore, *Chair*

David Anding
Wolfgang Baer
Stephen Baumgartner
Anthony Beverina
Paul Birkel
Dan Brockway
Greg Chisholm
William Cornette
Art Cortez
Jean Crowley
Rosemary Enright
Paul Foley

Brian F. Goldiez
James Hammond
Suraiya Haque
Louis Hembree
Donald Hook
John R. Hummel
Robert Jacober
Ronald S. Klasky
Todd Liddiard
Farid Mamaghani
David McKeeby
Gregory L. Mealy

Dale D. Miller
Russ Moulton
Kimberly Neff
David R. Pratt
Robert A. Reynolds
Richard Schaffer
Eric O. Schmidt
Graham Shanks
Michael Sieverding
Ralph M. Toms
Gregory Weidman
David A. Whitney

Field Instrumentation Working Group—Protocols Subgroup

Mark Hoptiak, *Chair*

Bruce Clay
Paul Cleveland
Gerald Lucha

Dana Patterson
Wes Potterton
Michael Rothrock

John Schockley
Jack Sherwood
Robert Sottolare

The following persons were on the balloting committee:

John B. Abernathy
Uhi Bernstein
Christina L. Bouwens
J. Joseph Brann
Donald Brutzman
Michael Burke
Frank Carr
Dominic Colonna
Kirk Dunkelberger
Jean-Pierre Faye
William J. Frederic
Rhonda Lea Freeman
Milton L. Fulghum, Jr.
Lawrence H. Goldberg
Brian F. Goldiez

Paul Gustavson
Frank R. Hill
Eric Hoffman
Mark Hoptiak
James Keenan
Robert Kerchner
Gerald Lucha
Peter McCarthy
Larry Bruce McDonald
David R. McKeeby
Robert E. Murray
Thomas Neuberger
Michael J. O'Connor
George Moody Parsons

David E. Poole
David R. Pratt
J. Mark Pullen
Randy Saunders
Joseph P. Schimming
Camile Schumacher
Steven M. Sheasby
Jack H. Sheehan
Randall K. Standridge
Gerald J. Stueve
James Totten
William V. Tucker
Greg J. Unangst
Gary L. Waag
Michael D. Wesdell

When the IEEE-SA Standards Board approved this standard on 19 March 1998, it had the following membership:

Richard J. Holleman, *Chair*

Donald N. Heirman, *Vice Chair*

Judith Gorman, *Secretary*

Satish K. Aggarwal
Clyde R. Camp
James T. Carlo
Gary R. Engmann
Harold E. Epstein
Jay Forster*
Thomas F. Garrity
Ruben D. Garzon

James H. Gurney
Jim D. Isaak
Lowell G. Johnson
Robert Kennelly
E. G. "Al" Kiener
Joseph L. Koepfinger*
Stephen R. Lambert
Jim Logothetis
Donald C. Loughry

L. Bruce McClung
Louis-François Pau
Ronald C. Petersen
Gerald H. Peterson
John B. Posey
Gary S. Robinson
Hans E. Weinrich
Donald W. Zipse

*Member Emeritus

Contents

1. Overview.....	1
1.1 Scope.....	1
1.2 Terminology.....	1
1.3 Key concepts.....	2
1.3.1 through 1.3.3	2
1.3.4 Functional requirements for DIS	2
2. References.....	4
3. Definitions, acronyms, and abbreviations.....	4
3.1 Definitions.....	4
3.2 Acronyms and abbreviations.....	5
4. General requirements	6
4.1 through 4.5.1.2	6
4.5.2 Entity information/interaction.....	8
4.5.3 Warfare	14
4.5.4 Logistics.....	14
4.5.5 Simulation management.....	14
4.5.6 Distributed emission regeneration	14
4.5.7 Radio communications.....	19
4.5.8 Entity Management.....	21
4.5.9 Minefield.....	34
4.5.10 Synthetic environment	39
4.5.11 Simulation management with reliability.....	46
5. Detailed requirements	69
5.1 through 5.1.3.2	69
5.1.4 Symbolic names	69
5.2 Basic data types and records	72
5.2.1 through 5.2.34	72
5.2.35 Acoustic Emitter System record	72
5.2.36 Aggregate Identifier record.....	73
5.2.37 Aggregate Marking record.....	73
5.2.38 Aggregate Type record	74
5.2.39 Beam Data record	74
5.2.40 Environment record	75
5.2.41 Environment Type record	76
5.2.42 Fundamental Operational Data record	77
5.2.43 Grid Axis Descriptor record.....	78
5.2.44 Grid Data record	80
5.2.45 IFF/ATC/NAVAIDS Fundamental Parameter Data record.....	82
5.2.46 Intercom Communications Parameters record.....	82
5.2.47 Layer Header record	83
5.2.48 Linear Segment Parameter record.....	83
5.2.49 Named Location Identification record	85
5.2.50 Object Appearance record.....	85

5.2.51	Object Identifier record.....	85
5.2.52	Object Type record	86
5.2.53	Propulsion System Data record.....	86
5.2.54	Record Query Specification record.....	87
5.2.55	Record Specification record.....	87
5.2.56	Relationship record	88
5.2.57	Secondary Operational Data record	88
5.2.58	System Identifier record.....	89
5.2.59	UA Fundamental Parameter Data record	89
5.2.60	Vectoring Nozzle System Data record.....	90
5.3	PDU's for DIS	91
5.3.1	Alignment	91
5.3.2	DIS PDU's.....	91
5.3.3	Entity Information/Interaction protocol family	92
5.3.4	Warfare protocol family.....	97
5.3.5	Logistics protocol family	97
5.3.6	Simulation Management protocol family	97
5.3.7	Distributed Emission Regeneration protocol family.....	97
5.3.8	Radio Communications protocol (RCP) family.....	105
5.3.9	Entity Management protocol family	110
5.3.10	Minefield protocol family	120
5.3.11	Synthetic Environment protocol family.....	131
5.3.12	Simulation Management with Reliability protocol family	142
6.	Non-Real Time protocol	165
6.1	Introduction.....	165
6.2	Definitions.....	165
6.3	Time management schemes	166
6.4	Data representation	166
6.5	Reliability.....	167
6.6	Time management services.....	167
6.6.1	Join Exercise Request service	167
6.6.2	Set Time Parameters service	169
6.6.3	Time Advance Report service.....	171
6.6.4	Time Advance Request service.....	172
6.6.5	Time Advance Grant service	173
6.6.6	Resign Exercise Request service	174
7.	Live Entity (LE) Information/Interaction protocol	176
7.1	Architectural principles.....	176
7.2	Basic records.....	177
7.2.1	LE PDU Header record.....	177
7.2.2	Relative World Coordinates record	178
7.2.3	Position Error record.....	179
7.2.4	Orientation Error record.....	179
7.3	LE Information/Interaction protocol family	179
7.3.1	TSPI PDU	180
7.3.2	Appearance PDU	184
7.3.3	Articulated Parts PDU.....	188
7.3.4	LE Fire PDU	190
7.3.5	LE Detonation PDU.....	195

Annexes A, B, and C (normative) (See annexes in IEEE Std 1278.1-1995).....	201
--	-----

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 seven six 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, acronyms, and abbreviations 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. Clause 6 contains the definition of a protocol specifically for applications operating in non-real time. Clause 7 is a stand-alone, self-contained clause that contains both the requirements and PDU definitions for use by live entities participating in a DIS exercise.

1.1 Scope

This standard is part of a set of standards and recommended practices for DIS applications. Each standard and recommended practice in the set describes one or more of the elements that constitute the DIS environment. As a whole, the set of standards and recommended practices will define an interoperable simulated engagement 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 are organized within specified domains called protocol families. The protocol families included in this standard include Entity Information/Interaction, Warfare, Logistics, Simulation Management, Distributed Emission Regeneration, Radio Communications, Entity Management, Minefield, Synthetic Environment, Simulation Management with Reliability, Live Entity Information/Interaction, and Non-Real Time.

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 of 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 through 1.3.3

See 1.3.1 through 1.3.3 in IEEE Std 1278.1-1995, IEEE Standard for Distributive Interactive Simulation—Application Protocols.¹

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) Simulation management
- g) Synthetic environment
- h) Entity management
- i) Minefield
- j) Live entity information/interaction
- k) Non-real time

All the above listed requirements are supported by this standard. A brief description of these areas follows.

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. An entity 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 elastic and in-elastic collisions. If 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

¹IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.

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.

Repair and resupply logistic services are modeled in a simulation exercise by means of the logistics 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. Audio communication includes both radio and intercom communications.

Representation of lasers, active electromagnetic emissions, and acoustic emissions, including active counter-measures are essential in certain simulation exercises. Entities with emitters simulate their emitter and output real-time operational parameters. Receiving entities that have receivers can then regenerate the transmitted signal based upon the simulated emitter output data and stored database information. Each receiving entity is responsible for determining whether the emission is detectable. If so, the receiving entity will use the emission data to appropriately influence its detection equipment or simulation of that equipment. In addition, support is included to communicate information for functions such as cooperative interrogator friend or foe (IFF) systems, air traffic control (ATC) beacon and transponder systems, and navigational aid systems (NAVAIDS).

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. A network manager would also have knowledge of host computers on the network, including their physical locations and network addresses. Analysis of network performance would be performed by the network manager. Management of the overall simulation exercise is also needed. Functions of simulation management include starting, restarting, pausing, and stopping of an exercise; creating and removing entities from an exercise; and collecting and distributing data with simulation applications. This standard includes protocols for the best effort and reliable communication of these exercise management messages.

For simulation entities to participate meaningfully in the same exercise, they must have access to the same synthetic 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, including engineering effects such as the construction of a bridge, berm, or building; weapons effects, which could destroy objects created via engineering effects and 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 engagements, as well as day engagements, should be simulated.

As the size of DIS exercises continued to grow, mechanisms were developed to allow the aggregation or grouping of entities during an exercise and for the reporting of the state of the group or aggregate in place of the states of the individual entities. Specific protocols are contained in this standard that support these capabilities and the capability to transfer the control of an entity from one simulation application to another.

Minefields are a specific type of DIS entity that are encountered in certain exercises. A protocol family has been specified in this standard to address the exchange of information about minefields and the mines contained therein. This exchange can be executed in either a heartbeat mode or in a query response protocol mode.

This standard supports application protocols for field instrumentation applications associated with live participants on instrumented ranges operating within the DIS environment and interacting with simulated and real entities. Limited communication bandwidth, volume of data, methods of data transmission, and type of data transmitted are the principal drivers for the specific set of protocols contained in this standard for use by live entities.

Generally, since most DIS exercises operate with a human in the loop, simulation time advances at the same rate as real-world time advances. As the scope of exercises using DIS expands, advancing time at a rate other than the rate at which real-world time advances is sometimes needed. This standard contains a specific protocol that allows simulation time to advance at a rate other than the rate at which real-world time advances. This protocol does not introduce any new messages into this standard, but rather specifies how existing messages should be used to support the needs of non-real time applications.

2. References

See Clause 2 and Annex C in IEEE Std 1278.1-1995.

3. Definitions, acronyms, and abbreviations

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

3.1 Definitions

The following definitions are in addition to the definitions in 3.1.1 through 3.1.37 in IEEE Std 1278.1-1995.

3.1.1 aggregate (unit): A group of entities or a group of other aggregates. The substitution of the word “unit” is used to avoid phrases like “aggregate aggregate.”

3.1.2 aggregation: The process of changing the resolution of an aggregate to represent it in less detail.

3.1.3 areal object: A synthetic environment object that is geometrically anchored to the terrain with a set of at least three points that come to a closure.

3.1.4 disaggregation: The process of changing the resolution of an aggregate to represent it in more detail.

3.1.5 linear object: A synthetic environment object that is geometrically anchored to the terrain with one point and has a segment size and orientation.

3.1.6 point object: A synthetic environment object that is geometrically anchored to the terrain with a single point.

3.2 Acronyms and abbreviations

The following acronyms and abbreviations are used in this standard:

AEDB	Active Emissions Database
ANDB	Additional Narrowband Database
APA	Additional Passive Activity
ATC	Air Traffic Control
BRG	Bearing
CADB	Common Acoustic Database
CGF	Computer-Generated Forces
CIR	Collision Intermediate Result
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 Protocols for Distributed Interactive Simulation Applications</i> . This document is frequently updated.
EM	Electromagnetic
ERP	Effective Radiated Power
GED	Group Entity Description
ID	Identification
IFF	Interrogator Friend or Foe
IR	Infrared
NACK	Negative Acknowledgment
NAVAIDS	Navigational Aids
NDA	Nozzle Deflection Angle
PDU	Protocol Data Unit
PgRF	Pulse group Repetition Frequency
QRP	Query Response Protocol

RCP	Radio Communications Protocol
RNG	Range
RPM	Revolutions Per Minute
SA	Simulation Application
SEDB	Special Event Database
SEES	Supplemental Emissions/Entity State
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
TI	Transaction Initiator
TR	Transaction Respondent
TSPI	Time Space Position Information
UA	Underwater Acoustic
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 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 5.1.4.

4.1 through 4.5.1.2

See 4.1 through 4.5.1.2 in IEEE Std 1278.1-1995.

4.5.1.3 Types and families of PDUs

This standard defines 40 new PDUs and a specific application protocol that utilizes existing PDUs. The PDUs and the specific application protocol in this standard are in addition to the 27 PDUs specified in IEEE Std 1278.1-1995. The consolidated list of PDUs and the associated protocol families is as follows:

- a) Entity information/interaction (defined in Clauses 4 and 5)
 - 1) Entity State PDU
 - 2) Collision PDU
 - 3) Collision-Elastic PDU (new)
 - 4) Entity State Update PDU (new)

- b) Warfare (defined in Clauses 4 and 5)
 - 1) Fire PDU
 - 2) Detonation PDU
- c) Logistics (defined in Clauses 4 and 5)
 - 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 (defined in Clauses 4 and 5)
 - 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 (defined in Clauses 4 and 5)
 - 1) Electromagnetic Emission PDU
 - 2) Designator PDU
 - 3) Underwater Acoustic (UA) PDU (new)
 - 4) IFF/ATC/NAVAIDS PDU (new)
 - 5) Supplemental Emission/Entity State (SEES) PDU (new)
- f) Radio Communications (defined in Clauses 4 and 5)
 - 1) Transmitter PDU
 - 2) Signal PDU
 - 3) Receiver PDU
 - 4) Intercom Signal PDU (new)
 - 5) Intercom Control PDU (new)
- g) Entity Management (new) (defined in Clauses 4 and 5)
 - 1) Aggregate State PDU (new)
 - 2) IsGroupOf PDU (new)
 - 3) Transfer Control Request PDU (new)
 - 4) IsPartOf PDU (new)
- h) Minefield (new) (defined in Clauses 4 and 5)
 - 1) Minefield State PDU (new)
 - 2) Minefield Query PDU (new)
 - 3) Minefield Data PDU (new)
 - 4) Minefield Response Negative Acknowledgment (NACK) PDU (new)
- i) Synthetic Environment (new) (defined in Clauses 4 and 5)
 - 1) Environmental Process PDU (new)
 - 2) Gridded Data PDU (new)
 - 3) Point Object State PDU (new)
 - 4) Linear Object State PDU (new)
 - 5) Areal Object State PDU (new)
- j) Simulation Management with Reliability (new) (defined in Clauses 4 and 5)
 - 1) Create-Entity-R PDU (new)
 - 2) Remove-Entity-R PDU (new)

- 3) Start/Resume-R PDU (new)
- 4) Stop/Freeze-R PDU (new)
- 5) Acknowledge-R PDU (new)
- 6) Action Request-R PDU (new)
- 7) Action Response-R PDU (new)
- 8) Data Query-R PDU (new)
- 9) Set Data-R PDU (new)
- 10) Data-R PDU (new)
- 11) Event Report-R PDU (new)
- 12) Comment-R Message PDU (new)
- 13) Record Query-R PDU (new)
- 14) Set Record-R PDU (new)
- 15) Record-R PDU (new)
- k) Live Entity (LE) (new) (defined in Clause 7)
 - 1) Time Space Position Information (TSPI) PDU (new)
 - 2) Appearance PDU (new)
 - 3) Articulated Parts PDU (new)
 - 4) LE Fire PDU (new)
 - 5) LE Detonation PDU (New)
- l) Non-Real Time protocol (new) (defined in Clause 6)

PDU type codes and protocol family codes in the range of 129 through 255 have been reserved for experimental purposes.

4.5.1.4 through 4.5.1.5

See 4.5.1.4 through 4.5.1.5 in IEEE Std 1278.1-1995.

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 either an Entity State PDU (see also 5.3.3.1) or an Entity State Update PDU (see also 5.3.3.4). 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) or a Collision-Elastic PDU (see also 5.3.3.3).

4.5.2.1 Entity state PDU

See 4.5.2.1 in IEEE Std 1278.1-1995.

4.5.2.2 Collision PDU

See 4.5.2.2 in IEEE Std 1278.1-1995.

4.5.2.3 Collision-Elastic PDU

The Collision-Elastic PDU is a member of a commercially available set of Newtonian Protocols and in that protocol set is named the Newtonian Collision PDU. The Newtonian Protocol provides a mechanism for introducing high fidelity interactions in DIS exercises. The Collision-Elastic PDU is used to communicate information about an elastic collision between two simulated entities that accounts for

- a) Linear and rotational momentum transfer
- b) Variable elasticity
- c) Momentum transfer that is dependent on surface orientation

Each Collision-Elastic PDU describes the information required to compute half of a high fidelity collision. The Collision-Elastic PDU can be a complete replacement for the Collision PDU (see 4.5.2.2 and 5.3.3.2) by setting to zero all fields after the Location of Impact field in the Collision-Elastic PDU (see Table 91).

4.5.2.3.1 Collision-elastic event initiation

The following shall constitute a collision detection:

- Receipt of a Collision-Elastic PDU from an entity with whom a state of collision does not presently exist.
- Detection of a bounding volume intersection with an entity with whom a state of collision does not presently exist.

A state of collision is any state other than the Rest state in the finite state machine described in 4.5.2.3.8.

4.5.2.3.2 Collision-elastic event termination

Collision events may be terminated in the following ways:

- An expected Collision-Elastic PDU response is not received after a time period of COLLISION_ELASTIC_TO.
- After receiving a Collision-Elastic PDU and after the collision is simulated, the bounding volumes of the two entities are separated.

If the bounding volumes continue to intersect, entities are entitled to move so that they do not intersect in order to reset the collision finite state machine (see 4.5.2.3.8).

4.5.2.3.3 Algorithm for collision-elastic computation

- Compute the contact velocity $\vec{v}_c$, which is the local velocity of the entities at the point of contact.

$$\vec{v}_{1c} = \vec{v}_1 + \vec{\omega}_1 \times \vec{r}_1$$

where

$\vec{v}$ is the linear velocity of the entity,

$\vec{\omega}$ is the angular velocity of the entity, and

$\vec{r}$ is the moment arm generated from the arbitrated location of impact.

All parameters are in world coordinates.

- Transform the own vehicle inverse tensor of inertia into world coordinates in order to represent it appropriately for use in the collision equations. (If a vector $\vec{s}$ in body coordinates is represented in world coordinates via the transformation $\underline{T}\vec{s}$, then $\underline{T}\underline{I}^{-1}\underline{T}^T$ is the representation of the inverse tensor of inertia in world coordinates.) Compute a matrix, which is denoted by $\underline{Q}_1$ called the Collision

Intermediate Result by $\underline{L}_{\vec{r}_1}^T \underline{I}_1^{-1} \underline{L}_{\vec{r}_1}$, where $\underline{L}_{\vec{r}_1} = [\vec{r}_1 X] = \begin{bmatrix} 0 & -r_1[3] & r_1[2] \\ r_1[3] & 0 & -r_1[1] \\ -r_1[2] & r_1[1] & 0 \end{bmatrix}$, i.e., the matrix

operator that results in computing the cross product with $\vec{r}_1$.

- c) Send η_1 the coefficient of restitution, m_1 the mass, l_1 the location of the point of contact, $\hat{n}_1$ the unit surface normal, $\underline{Q}_1$ the Collision Intermediate Result, and $\vec{v}_{1c}$ the contact velocity in a Collision-Elastic PDU. Receive η_2 the coefficient of restitution, m_2 the mass, l_2 the location of the point of contact, $\hat{n}_2$ the unit surface normal, $\underline{Q}_2$ the Collision Intermediate Result, and $\vec{v}_{2c}$ the contact velocity in a Collision-Elastic PDU.
- d) Arbitrate the data as described in 4.5.2.3.7.
- e) Solve for the momentum transfer given by $\kappa = -\frac{(1 + \eta)(\vec{v}_{c1} - \vec{v}_{c2}) \times \hat{n}}{\frac{1}{m_1} + \frac{1}{m_2} + \hat{n}^T \underline{Q}_1 \hat{n} + \hat{n}^T \underline{Q}_2 \hat{n}}$, where η and $\hat{n}$ are the arbitrated values of the coefficient of restitution and unit surface normal, respectively.
- f) Substitute the momentum transfer into the following velocity update equations: $\vec{v}'_1 = \vec{v}_1 + \hat{n} \kappa / m_1$ and $\vec{w}'_1 = \vec{w}_1 + \underline{I}_1^{-1}(\vec{r} \times \hat{n}) \kappa$ where I is the tensor of the own vehicle in world coordinates.

4.5.2.3.4 Information contained in a Collision-Elastic PDU

The Collision-Elastic PDU shall contain the following information:

- a) Identity of the entity that issued the Collision-Elastic PDU
- b) Identification of the entity with which the issuing entity collided (If this information 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) Velocity of the issuing entity at the point of contact, to be distinguished from the velocity of the center of mass
- e) Mass of the issuing entity
- f) Location of impact in entity coordinates of the entity with which the issuing entity collided
- g) Six-member array of the upper triangular part of the symmetric matrix called the Collision Intermediate Result (CIR). The members of the array are given by CIR_{XX} , CIR_{XY} , CIR_{XZ} , CIR_{YY} , CIR_{YZ} , and CIR_{ZZ} . The CIR is represented by the matrix product $\underline{L}_{\vec{r}_1}^T \underline{I}_1^{-1} \underline{L}_{\vec{r}_1}$, where $\underline{I}_1$ is the tensor of inertia and $\underline{L}_{\vec{r}_1}^T$ is given by $\underline{L}_{\vec{r}_1} = [\vec{r}_1 X] = \begin{bmatrix} 0 & -r_1[3] & r_1[2] \\ r_1[3] & 0 & -r_1[1] \\ -r_1[2] & r_1[1] & 0 \end{bmatrix}$.
- h) Unit surface normal at the point of contact in world coordinates
- i) Coefficient of restitution, which represents a tunable parameter in the collision equation

4.5.2.3.5 Issuance of the Collision-Elastic PDU

A Collision-Elastic PDU shall be issued by an entity when a collision is detected between the issuing entity and another entity with whom the issuing entity is not already engaged in a collision simulation. Only after termination of the prior collision simulation may the next collision simulation begin. See 4.5.2.3.1 for the conditions defining a collision detection. An entity that receives a Collision-Elastic PDU indicating another entity has collided with it without first detecting the collision shall issue a Collision-Elastic PDU naming the entity that issued the first Collision-Elastic PDU. Steps should be taken to ensure that, if and when it subsequently detects the same collision event, it does not generate a Collision-Elastic PDU to report it. When a Newtonian simulation application receives a Collision-Elastic PDU naming an entity it simulates as the other party involved in a Newtonian collision after reporting the same collision event, it shall not send another Collision-Elastic PDU in response. A Newtonian collision simulation application shall always issue

one, and only one, Collision-Elastic 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-Elastic PDU shall never be issued in response to the Collision PDU specified in 4.5.2.2 of IEEE Std 1278.1-1995.

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

4.5.2.3.6 Receipt of the Collision-Elastic PDU

Entities that are not prepared to engage in a Newtonian collision simulation shall ignore the Collision-Elastic PDU.

Entities capable of performing Newtonian collision simulations shall simulate the collision upon receipt of the PDU. If a Collision-Elastic PDU has not already been sent, then the recipient of the Collision-Elastic PDU shall respond by issuing a Collision-Elastic PDU. The simulation shall then wait for the conditions necessary to terminate a collision. See 4.5.2.3.2 for the conditions defining the termination of a collision event.

4.5.2.3.7 Arbitration of collision parameters

Both the coefficient of restitution η and the unit surface normal $\hat{n}$ shall be arbitrated using the following pre-determined arbitration functions:

$$\hat{n} = \frac{\hat{n}_1 - \hat{n}_2}{\|\hat{n}_1 - \hat{n}_2\|}$$

$$\eta = \min(\eta_1, \eta_2)$$

The location of impact shall be arbitrated by converting the location of impact in the received Collision-Elastic PDU into world coordinates. The arithmetic mean shall then be computed. This location of impact shall then be used to recompute the moment arm $\vec{r}$ used in the velocity update equations.

It is recommended that collisions always have elasticity to result in the ultimate separation of the collision entities, thereby guaranteeing that the collision detection finite state machine terminates.

Often the receiver of a Collision-Elastic PDU cannot detect a collision via geometric considerations. The variables necessary to compute a collision may not be accessible. The appropriate variable needs to be synthesized for the response. The recommended approach is to accept all data that would otherwise be arbitrated and format them to be sent in the Collision-Elastic PDU response as follows:

- a) Coefficient of restitution shall be retransmitted as it is received.
- b) Unit surface normal, which is received in world coordinates, shall be negated and retransmitted.
- c) Location of impact is received in own body coordinates. This result shall be translated to the body coordinate of the remote vehicle and retransmitted.
- d) Collision intermediate result and the contact velocity both rely on the moment arm, which shall be computed from the location of impact.

4.5.2.3.8 State information for Newtonian collision simulations

The following paragraphs describe the different states and transitions for Newtonian collision simulations.

Each entity may be in one of four states:

- a) *Rest state.* A receiving entity is in the Rest state before interacting with a given entity.
- b) *PDU Sent state.* An entity is in the PDU Sent state when a collision is detected without having received a Collision-Elastic PDU.
- c) *Simulate state.* This transitory state occurs when a Collision-Elastic PDU is received. Computation and distribution of new kinematic variables occurs here. Damage models are also computed while in this state.
- d) *Wait state.* The kinematics of the collision have been simulated and the entity is waiting for the conditions that terminate the collision. One may choose other mechanisms than the collision results to terminate the collision.

The behavior of the Newtonian collision simulation is shown in Figure 21. The associated state transitions are described in Table 60.

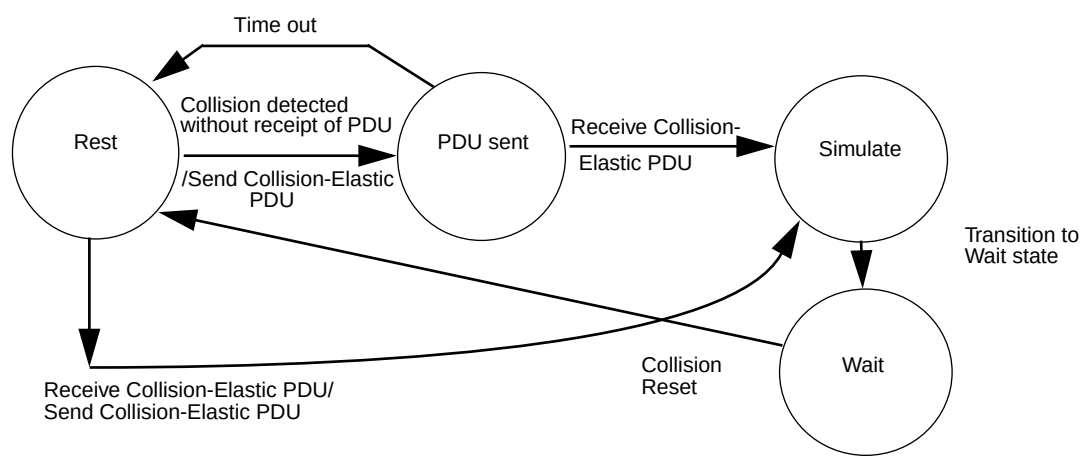


Figure 21—Newtonian collision finite state machine

Table 60—Newtonian collision simulation state transition

Transition	Conditions and actions
Collision detected without receipt of Collision-Elastic PDU/Send Collision-Elastic PDU	This transition is only relevant in the Rest state. When this condition is relevant, a Collision-Elastic PDU shall be issued. The entity shall then proceed to the PDU Sent state.
Receive Collision-Elastic PDU	When a Collision-Elastic PDU is received while the entity is in the PDU Sent state, the entity shall transition to the Simulate state.
Time Out	This transition is only valid in the PDU Sent state. If the entity is in this state for a time greater than COLLISION_ELASTIC_TO, then the finite state machine shall reset and transition to the Rest state.
Transition to Wait state	This transition occurs immediately after being in the transitional Simulate state. No condition is required to transition to the Wait state from the Simulate state.
Collision Reset	This transition is only relevant in the Wait state. When the entities have separated, the entity shall return to the Rest state.
Receive Collision-Elastic PDU/ Send Collision-Elastic PDU	When a Collision-Elastic PDU is received in the Rest state, a Collision-Elastic PDU shall be issued. The entity shall then proceed to the Simulate state.

4.5.2.4 Entity State Update PDU

The Entity State Update PDU is a network bandwidth-reducing extension of the basic Entity State PDU and may be used to communicate specific nonstatic information about an entity's state. The Entity State Update PDU should not serve as a complete replacement for the Entity State PDUs (see 4.5.2.1) since the update PDU does not convey an entity's static and nearly static information.

4.5.2.4.1 Information contained in an Entity State Update PDU

The Entity State Update PDU shall contain the following information:

- a) Identification of the entity that issued the PDU
- b) Information about the location of the entity in the simulated world and its orientation, including
 - 1) Velocity
 - 2) Location with respect to the world
 - 3) Orientation
- c) Information required for representation of the entity's appearance, including
 - 1) Number of articulation parameters and the parameter values to represent orientation of articulated parts
 - 2) Appearance of the entity (e.g., normal, smoking, on fire, producing a dust cloud)
 - 3) Presence (including types and numbers) of attached parts or stores

4.5.2.4.2 Issuance of the Entity State Update PDU

A simulation shall issue an Entity State Update 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 the dead reckoning algorithm, see Annex B, specified in the Entity State PDU) exceeds a predetermined threshold (see 4.5.2.1.2 in IEEE Std 1278.1-1995 concerning threshold values). This threshold includes changes in position/orientation information and articulated part parameter information (see A.2.1.6).
- b) A change in the entity's appearance occurs.
- c) A predetermined length of real-world time has elapsed since the issuing of the last Entity State or Entity State Update PDU. This value and associated tolerance may be established at the start of an 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 or Entity State Update PDU shall be issued to indicate that the entity is in the Deactivated state.

The Entity State Update 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.

4.5.2.4.3 Receipt of the Entity State Update PDU

Upon receipt of an Entity State Update 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 (if present) 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 or Entity State Update 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 Update PDU issuance interval [see 4.5.2.4.2 item c)] multiplied by HRT_BEAT_MPLIER.

4.5.3 Warfare

See 4.5.3 in IEEE Std 1278.1-1995.

4.5.4 Logistics

See 4.5.4 in IEEE Std 1278.1-1995.

4.5.5 Simulation management

The content and 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. See 4.5.11 for the requirements for Simulation Management PDUs, based on the PDUs in this subclause, that incorporate a mechanism to allow the specification of the application-level communications reliability service to be used for the PDU exchange.

4.5.5.1 through 4.5.5.6.3

See 4.5.5.1 through 4.5.5.6.3 in IEEE Std 1278.1-1995.

4.5.6 Distributed emission regeneration

The content and procedure for the use of Distributed Emission Regeneration (DER) in a DIS exercise is established in 4.5.6.1 through 4.5.6.6.

4.5.6.1 Distributed emission regeneration approach

See 4.5.6.1 in IEEE Std 1278.1-1995.

4.5.6.2 Electromagnetic Emission PDU

See 4.5.6.2 in IEEE Std 1278.1-1995.

4.5.6.3 Designator PDU

See 4.5.6.3 in IEEE Std 1278.1-1995.

4.5.6.4 UA PDU

The UA PDU shall be used to communicate UA active emissions (intentional emissions) and passive signature (unintentional emissions) information.

4.5.6.4.1 Information contained in the UA PDU

The UA PDU shall contain the following information:

- a) Identification of the emitting entity
- b) Identification of the event
- c) Identification of the type of update information in the UA PDU

- 1) A state update shall provide a full description of the emission system(s) and passive signature information identified in the UA PDU
- 2) A change data update shall provide a method of allowing changes of UA emitter state or passive signature to be communicated between state updates
- d) Information required to define the entity's passive signature, including
 - 1) Passive parameter index
 - 2) Propulsion plant configuration
 - 3) Number of shafts
 - 4) Current and ordered shaft speed
 - 5) Shaft speed rate of change
 - 6) Number of Additional Passive Activities (APAs)
 - 7) APA parameter index
 - 8) APA value
- e) Information required to identify the active emission platform characteristics, including
 - 1) Number of UA emitter systems for which information is being provided in the UA PDU
 - 2) Information for one or more UA emitter systems, including
 - i) Length of the emitter system data for each system
 - ii) Number of main beams for each system
 - iii) Acoustic emitter system information, which includes the acoustic emitter system name, function of the system and acoustic identification number
 - iv) Location of the UA emitter with respect to the entity
 - v) Information for one or more acoustic beams that the system has, including
 - Length of the beam data
 - Beam identification number for each beam
 - Fundamental parametric data used to define the entity's active emissions. This field defines the active emission parameter index, beam scan pattern, orientation, and beamwidth, which can vary dynamically during system operation

4.5.6.4.2 Issuance of the UA PDU

Entities that simulate UA noise sources will output acoustic noise parameters that, in conjunction with database information, describe intentional and unintentional UA emissions. The UA PDU shall be issued in the following instances:

- a) State update
 - 1) An Entity State PDU for the associated entity is first issued and the entity is generating passive signature information or has at least one acoustic emitter that is presently active.
 - 2) A predetermined length of real-world time has elapsed since issuing the last UA PDU. This value and associated tolerance may be established at exercise start or during the exercise by means of the Simulation Management PDUs. If no value is established, the default value shall be UA_STATE_UPDATE_HRT_BEAT_TIMER. A default tolerance of $\pm 10\%$ shall apply to this value.
- b) Changed data update: Emitter operational parameters or entity configuration data contained in the UA PDU fields change. The UA PDU for a changed data update shall contain only data for those parameters that exceed the defined thresholds. These thresholds may be established at exercise start or during the exercise by means of the Simulation Management PDUs. If no threshold value is established, the default threshold values specified below shall be used. The parameters that can initiate a changed data update are
 - 1) Propulsion plant configuration change, i.e., any change in the entity propulsion plant configuration.
 - 2) Shaft revolutions per minute (RPM) change. Unless otherwise established, the default threshold shall be UA_SRPM_CHANGE_DELTA.
 - 3) Shaft RPM rate of change. Unless otherwise established, the default threshold shall be UA_SRPM_ROC_CHANGE_DELTA.

- 4) APA additions, deletions, or parametric changes to current activity.
- 5) A change in the number of active acoustic emitter systems.
- 6) A change in the number of active main beams per system.
- 7) Location with respect to the entity (e.g., a towed active emitter) changes by UA_LOCATION_CHANGE_DELTA.
- 8) Fundamental Data Parameters change, i.e., a change in the Active Emission Parameter Index values (e.g., pulse type, pulse duration, pulse interval, source power, pulse frequency content, pulse beam pattern), Scan Pattern (if applicable), or main beam steering by UA_ORIENTATION_CHANGE_DELTA in azimuth or depression/elevation.

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

4.5.6.4.3 Receipt of the UA PDU

Upon receipt of an UA PDU (and corresponding Entity State PDU), the receiving UA simulations shall determine if the emission or signature is detectable and use the information in the UA PDU and predefined databases to appropriately influence its UA detection equipment or simulation of that equipment in those simulations. The receiving simulation shall be responsible for simulating the environment and transmission effects (including reflections) from the noise source to the sensor(s). Environmental conditions will be provided by the Environmental Process PDU (see 4.5.10.1.1) and the Gridded Data PDU (see 4.5.10.1.2).

4.5.6.4.4 UA emission regeneration

Active and passive entity emissions and signatures shall be generated based on parameters stored in databases of the receiving UA simulations as enumerated by the UA PDU and based on the dynamic operational parameters included in the UA PDU.

4.5.6.5 IFF/ATC/NAVAIDS PDU

The IFF/ATC/NAVAIDS PDU shall be used to communicate information for functions such as cooperative IFF systems, ATC beacon and transponder systems, collision avoidance, and NAVAIDS systems.

4.5.6.5.1 Information contained in an IFF/ATC/NAVAIDS PDU

The IFF/ATC/NAVAIDS PDU shall contain at least the following information:

- a) Identification of the entity emitting IFF/ATC/NAVAIDS signals
- b) Event identification of the specific event
- c) Identification of the type of system emitting the signals
- d) Identification of the location of the emitting system with respect to the emitting entity
- e) Fundamental operational data for the system, reflecting at a minimum the enabling status of the system. The enabling status includes which modes of signals the system is capable of emitting and the settings for these modes (codes transmitted). The intent is to capture the state of switches and dials an operator might change during an exercise.
- f) Additional layers of information, beyond Layer 1, may be added in the IFF/ATC/NAVAIDS PDU indicating in greater detail the actual emission characteristics of the system. Each of these additional layers shall include the layer length and identity. Information concerning the following aspects of IFF/ATC/NAVAIDS systems would be communicated through these additional layers:
 - 1) Electromagnetic characteristics of a typical emission
 - 2) Rate at which signals are emitted
 - 3) Identification of which signals are responses to which previous signals

4.5.6.5.2 Issuance of the IFF/ATC/NAVAIDS PDU

The IFF/ATC/NAVAIDS PDU with at least Layer 1 information shall be issued when

- a) An Entity State PDU for the associated entity is first issued and IFF/ATC/NAVAIDS data exist for an associated interrogator or transponder for which at least one mode is presently on.
- b) A predetermined length of real-world time has elapsed since the issuing of the last IFF/ATC/NAVAIDS PDU. This Periodic Transmission Parameter value 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 IFF/ATC/NAVAIDS_DFLT.
- c) The value of any parameter represented in the Layer 1 information of the IFF/ATC/NAVAIDS PDU (excluding the header) has changed. When a value changes, the IFF/ATC/NAVAIDS PDU [containing the new value(s) and the Change Indicator of the Change/Options field in the System Identifier record indicating a change has occurred] shall be issued within a predetermined length of real-world time. This Change Latency Parameter value may be established at the start of the exercise or during the exercise by means of Simulation Management PDUs. If no value is established, a default value of IFF/ATC/NAVAIDS_CLAT shall be used. The next IFF/ATC/NAVAIDS PDU issued under item b) above (periodic update) shall also have the Change Indicator indicating a change has occurred. Thereafter, IFF/ATC/NAVAIDS PDUs issued shall have the Change Indicator indicating no change until another change occurs.
- d) More frequent issuance has been coordinated among the exercise participants.

Issuance of an IFF/ATC/NAVAIDS PDU is not required until at least one represented mode of IFF/ATC/NAVAIDS is on. However, once an IFF/ATC/NAVAIDS PDU is transmitted, it shall continue to be periodically transmitted regardless of the system status or mode status values.

All simulations representing IFF/ATC/NAVAIDS functions shall be capable of issuing IFF/ATC/NAVAIDS PDUs with Layer 1 information. It is recommended that the use of additional information layers be coordinated between the exercise participants prior to an exercise or by use of Simulation Management PDUs. When additional information layers are present, each layer shall be positioned within the IFF/ATC/NAVAIDS PDU in numerical order. When more than two information layers are specified, the appearance together of certain combinations of information layers may be inappropriate in the same IFF/ATC/NAVAIDS PDU. The inappropriate combinations will be defined as additional information layers are specified.

Issuance rules dependent on system type are defined in Section 8 of *Enumeration and Bit-Encoded Values for Use with Protocols for Distributed Interactive Simulation Applications* (EBV-DOC).²

The IFF/ATC/NAVAIDS PDU shall be issued using a best effort multicast communication service.

4.5.6.5.3 Receipt of the IFF/ATC/NAVAIDS PDU

Upon receipt of the IFF/ATC/NAVAIDS PDU, entities simulating the IFF/ATC/NAVAIDS functions shall use the information in Layer 1 of the IFF/ATC/NAVAIDS PDU to simulate the enabling status of the emitting system. If additional information layers are present, these layers may be used to represent the state of the system on the emitting entity during the period preceding the PDU timestamp. This information may be used to predict the present state of such a system.

The System Status On/Off (see Section 8 of EBV-DOC) value shall have precedence over the On/Off status value for Parameter 1–6 or the Modifier Parameters. If the System Status is Off, all Parameter 1–6 and Modifier Parameter modes shall be considered off. If the System Status is On, each Parameter 1–6 and Modifier Parameter mode shall be considered on or off depending on the On/Off status for that parameter.

²For information on this and other projects underway at the Defense Modeling, Simulation and Tactical Technology Information Analysis Center, check their web site at <http://dmsttiac.hq.iitri.com/>.

4.5.6.6 SEES PDU

The SEES PDU shall be used to communicate supplemental information concerning passive emissions (infrared and radar cross section) and active emissions (acoustic), along with additional information pertaining to propulsion systems contained within the entity.

4.5.6.6.1 Information contained in the SEES PDU

The SEES PDU shall contain the following information:

- a) Identification of the originating entity
- b) Infrared signature representation index
- c) Acoustic signature representation index
- d) Radar cross-section signature representation index
- e) Information required to define the entity's propulsion systems, including
 - 1) Number of propulsion systems
 - 2) Number of vectoring nozzle systems
 - 3) Propulsion system data
 - 4) Vectoring nozzle system data

4.5.6.6.2 Issuance of the SEES PDU

Entities that simulate infrared and surface acoustic signatures shall output parameters that, in conjunction with database information, describe their own infrared and acoustic emissions. Also, entities shall output parameters that in conjunction with database information, describe their own radar cross-section information. The SEES PDU shall be issued in the following instances:

- a) State update
 - 1) An Entity State PDU for the associated entity is first issued and the entity is generating supplemental emission/entity information.
 - 2) A predetermined length of real-world time has elapsed since issuing the last SEES PDU. This value and associated tolerance may be established at exercise start or during the exercise by means of the Simulation Management PDUs. If no value is established, the default value shall be SEES_STATE_UPDATE_DFLT. A tolerance of $\pm 10\%$ shall apply to this value.
- b) Changed data update. Propulsion or nozzle system parameters contained in the SEES PDU fields change more than a specified threshold, or the state of the entity's infrared, acoustic, or presented radar cross-section signature changes so that a new representation index must be issued. A SEES PDU for a changed data update shall contain only data for the parameters that exceed the defined thresholds. If a threshold is not defined, the default threshold values listed below shall be used. The parameters that can initiate a changed data update are
 - 1) Representation index change, i.e., any change in the value of the representation index (infrared, acoustic, or radar cross section).
 - 2) Power setting change. Unless otherwise specified by the user, the default threshold shall be SEES_PS_THRSH_DFLT.
 - 3) Engine RPM change. Unless otherwise specified by the user, the default threshold shall be SEES_RPM_DELTA_THRSH_DFLT.
 - 4) Nozzle deflection angle (NDA) change. Unless otherwise specified by the user, the default NDA threshold shall be SEES_NDA_THRSH_DFLT.

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

4.5.6.6.3 Receipt of the SEES PDU

Upon receipt of a SEES PDU (and the most recently received corresponding Entity State PDU), the receiving SEES simulations shall determine whether the signatures are detectable and use the information in the SEES PDU and predefined databases to appropriately influence infrared, acoustic, and/or radar detection equipment. The receiving simulation shall also be responsible for modeling any environmental and transmission effects between the signal source, entity, and sensor receiver. These environmental conditions shall be provided by the Environmental Process PDU (see 4.5.10.1.1), Gridded Data PDU (see 4.5.10.1.2), and the static environment.

4.5.7 Radio communications

The protocol for the simulation of radio communications, including intercom communications, in DIS is described in 4.5.7.1 through 4.5.7.6. The Radio Communications Protocol (RCP) supports the simulation of both audio and data transmission by radio and Tactical Data Links (TDLs). In the RCP, the content of the radio communication may be conveyed in its entirety in real-time or may be conveyed by reference to a pre-recorded database.

4.5.7.1 Radio communications approach

See 4.5.7.1 in IEEE Std 1278.1-1995.

4.5.7.2 Transmitter PDU

See 4.5.7.2 in IEEE Std 1278.1-1995.

4.5.7.3 Signal PDU

See 4.5.7.3 in IEEE Std 1278.1-1995.

4.5.7.4 Receiver PDU

See 4.5.7.4 in IEEE Std 1278.1-1995.

4.5.7.5 Intercom Signal PDU

The Intercom Signal PDU shall be used to convey the audio or digital data that is used to communicate between simulated intercom devices.

4.5.7.5.1 Information contained in the Intercom Signal PDU

The Intercom Signal PDU shall contain the content of an intercom transmission. This content may be digitized audio, binary data, or an index into a database that defines the data.

The Intercom Signal PDU shall contain the following information:

- a) Identification of the entity that is the source of the transmission
- b) Identification of the particular device that is transmitting
- c) Specification of the encoding scheme utilized. The encoding scheme shall be as specified in 4.5.7.3.1 item c).
- d) Specification of the type of TDL message included in the Intercom Signal PDU. The TDL type shall be as specified in 4.5.7.3.1 item d).
- e) Sample rate, in samples per second, for audio data and/or the data rate, in bits per second, for digital data

- f) Length of the data fields expressed in bits
- g) Number of individual audio/voice samples
- h) Audio/voice data

4.5.7.5.2 Issuance of the Intercom Signal PDU

An Intercom Signal PDU shall be issued whenever intercom audio or data are being transmitted on any line of the simulated intercom device.

The encoding class may change within the Intercom Signal PDUs of a single transmission; however, the encoding type for a particular communications device shall not change for the duration of a simulation exercise.

The Intercom Signal PDU shall be issued using a best effort multicast communication service.

4.5.7.5.3 Receipt of the Intercom Signal PDU

Upon receipt of the Intercom Signal PDU, the receiving intercom simulation application shall determine whether the issuing simulated intercom device is currently transmitting information on a channel that the receiving simulated intercom device is simulating. If so, the receiving intercom simulation application shall convey the voice or data to the appropriate local destinations.

4.5.7.6 Intercom Control PDU

The Intercom Control PDU shall be used to communicate the state of a particular intercom device, request an action of another intercom device, or respond to an action request.

4.5.7.6.1 Information contained in the Intercom Control PDU

The Intercom Control PDU shall contain the following information:

- a) Identification of the type of request being made by the Intercom Control PDU
- b) Identification of the type of communications channel that is being simulated
- c) Identification of the source of this Intercom Control PDU, including
 - 1) Identification of the entity associated with the communications device
 - 2) Identification of the communications device by sequence number
 - 3) Identification of the line number of the communications device by sequence number
- d) Transmit priority of the communications device on the line
- e) Transmit line state of the communications device on the line
- f) Identification of the type of Intercom Control PDU
- g) Identification of the master of the intercom channel, including
 - 1) Identification of the entity associated with the communications device
 - 2) Identification of the communications device by sequence number
 - 3) Identification of this communications device's intercom channel by sequence numberTogether, these three fields provide an intercom channel identification that is unique exercise-wide.
- h) Additional parameter information, including destination(s) of the Intercom Control PDU as required

4.5.7.6.2 Issuance of the Intercom Control PDU

An Intercom Control PDU shall be issued by an intercom simulation application when

- a) The simulated intercom device has a change in the transmit state or priority on a particular line or has any other change in state of which other simulated intercom devices need to be aware
- b) An action of another simulated intercom device is requested
- c) A response to a request of another simulated intercom device is made

The Intercom Control PDU shall be issued using a best effort multicast communication service.

4.5.7.6.3 Receipt of the Intercom Control PDU

Upon receipt of an Intercom Control PDU, the receiving intercom simulation application shall determine the type of Intercom Control PDU. If the PDU contains only transmit state information and the receiving simulated intercom device is currently involved in a simulation of the channel indicated in the Intercom Control PDU, the receiving intercom simulation application shall take appropriate actions to change its internal state based on its internal logical model. If the PDU contains a request, the receiving simulated intercom device shall determine whether it is the intended recipient of the PDU, perform the request if possible, and send an acknowledgment to the requestor if required. If the PDU contains an acknowledgment, the receiving simulated intercom device shall determine whether it is the intended recipient of the PDU and take appropriate actions based on the type of acknowledgment and the logic model.

4.5.8 Entity Management

Protocols for controlling entity aggregation activities, communicating the state of grouped entities, transferring the ownership of an entity, and requesting a hierarchical linkage of separately hosted simulation entities are defined in 4.5.8.1 through 4.5.8.4. The protocols of these four functional capabilities are not explicitly coupled, but rather provide alternative means for addressing several entity management needs within a DIS exercise.

4.5.8.1 Aggregation

The Aggregate State PDU in conjunction with three Simulation Management PDUs provides a capability for aggregating entities and communicating information about these aggregates of entities. It also provides a mechanism for participants in a DIS exercise to request the level at which these entities are represented.

The three Simulation Management PDUs (see 4.5.5) used with the Aggregate State PDU are the

- a) Action Request PDU
- b) Action Response PDU
- c) Event Report PDU

When these Simulation Management PDUs are used in the aggregation capability of the Entity Management protocol, they shall specify the Entity Management protocol in the Protocol Family field of the PDU header.

4.5.8.1.1 Aggregate State PDU

The Aggregate State PDU shall be used to communicate the state and other pertinent information about an aggregated unit.

4.5.8.1.1.1 Information contained in the Aggregate State PDU

The Aggregate State PDU shall contain the following information:

- a) Identification of the aggregate that issued the PDU
- b) Identification of the force to which the aggregate belongs
- c) State of the aggregate
- d) Type of the aggregate
- e) Formation of the aggregate
- f) Marking of the aggregate
- g) Aggregate unit information, including
 - 1) Size of the bounding volume

- 2) Orientation
- 3) Location of the center of mass
- 4) Linear velocity
- h) Number of DIS aggregates
- i) Number of DIS entities
- j) Number of silent aggregate types
- k) Number of silent entity types
- l) Identification for each subaggregate issuing Aggregate State PDUs
- m) Identification for each entity that is a part of the aggregate and that is also issuing Entity State PDUs
- n) Information about each type of silent subaggregate (not issuing Aggregate State PDUs), including
 - 1) Number of silent subaggregates of this type
 - 2) Type of the aggregate
- o) Information about each type of silent entity (not issuing Entity State PDUs), including
 - 1) Number of silent entities of this type
 - 2) Number of appearance records
 - 3) Type of the entity
 - 4) Appearance records for some of these entities
- p) Exercise-specific information

Simulation applications representing the aggregate must be careful not to exceed the maximum PDU size specified in IEEE Std 1278.2-1995 for the Aggregate State PDU.

4.5.8.1.1.2 Issuance of the Aggregate State PDU

The aggregate shall transmit an Aggregate State PDU when a predetermined length of real-world time has elapsed since the issuing of the last Aggregate State PDU. This value and associated tolerance may be established at the start of the exercise or during the exercise by means of the Simulation Management PDUs. If no value is established, the default value shall be AGG_HRT_BEAT_TIMER. A default tolerance of $\pm 10\%$ shall apply to this value. Each non-silent aggregate shall issue an Aggregate State PDU. A silent aggregate shall not issue an Aggregate State PDU when it is being represented in the Aggregate State PDU of a higher-level aggregate. Also, immediately after a unit is aggregated or disaggregated, the aggregate shall transmit an Aggregate State PDU to indicate the new state.

The Aggregate State PDU shall be issued using a best effort multicast communication service.

4.5.8.1.1.3 Receipt of the Aggregate State PDU

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

4.5.8.1.2 Action Request PDU

The Action Request PDU shall be used in the aggregation capability of the Entity Management protocol to request that a unit be represented at a different level.

4.5.8.1.2.1 Information contained in the Action Request PDU

The Action Request PDU shall contain the following information:

- a) Identification (ID) of the aggregate requested to change state (Receiving Entity ID)
- b) Identification of the entity, aggregate, or simulation requesting the change of state (Originating Entity ID)

- c) Identification number of the request to change state
- d) Identification of the specific action to be taken
- e) Data required for performing the requested action as follows:
 - 1) Requested state of the aggregate,
 - 2) Trigger used to determine this request,
 - 3) Detection range of the entity requesting the change of state,or
 - 4) Estimated time that the unit is requested to be in the new state.

4.5.8.1.2.2 Issuance of the Action Request PDU

The Action Request PDU shall be issued by any application that wants a unit to be represented at a different level and shall be issued using a best effort unicast or multicast communication service.

4.5.8.1.2.3 Receipt of the Action Request PDU

The Action Request PDU shall be received by the aggregate. If the aggregate receives conflicting requests, it is recommended, but not mandatory, that the unit be represented at the lowest level requested.

4.5.8.1.3 Action Response PDU

The Action Response PDU shall be used in the aggregation capability of the Entity Management protocol to communicate the status of the action requested in the Action Request PDU.

4.5.8.1.3.1 Information contained in the Action Response PDU

The Action Response PDU shall contain the following information:

- a) Identification of the entity, aggregate, or simulation that sent the Action Request PDU (Originating Entity ID)
- b) Identification of the aggregate requested to change state (Receiving Entity ID)
- c) Identification number of the original request to change state
- d) Status of the action request

4.5.8.1.3.2 Issuance of the Action Response PDU

The Action Response PDU shall be issued by the aggregate in response to an Action Request PDU and shall be issued using a best effort unicast or multicast communication service.

4.5.8.1.3.3 Receipt of the Action Response PDU

Upon receipt of the Action Response PDU, the application that sent the original request shall note that the Action Request PDU was received and the status of that request.

4.5.8.1.4 Event Report PDU

The Event Report PDU shall be used in the aggregation capability of the Entity Management protocol to communicate a change of state of an aggregate.

4.5.8.1.4.1 Information contained in the Event Report PDU

The Event Report PDU shall contain the following information:

- a) Identification of the aggregate issuing the PDU

- b) Identification of the simulation application to receive the Event Report PDU
- c) Identification of the type of change of state that has been requested
- d) Identification of the status of the completion of the request

4.5.8.1.4.2 Issuance of the Event Report PDU

The Event Report PDU shall be issued by the aggregate to warn other simulations of a change state request that may affect them. It may also be issued to warn other simulations that the simulation applications representing the aggregate want to change the state of a unit. The exercise participants must determine before the exercise which requested states should be reported. If a response is desired, then the exercise participants must also determine how long the aggregate will wait for that response. If no value is specified, the default shall be AGG_RESPONSE_DFLT. Typically, the aggregation of a unit is reported, while the disaggregation of a unit is not reported.

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

4.5.8.1.4.3 Receipt of the Event Report PDU

Upon receipt of the Event Report PDU, the application shall determine whether the requested change of state affects them. If the status of the change state request is pending, then other simulations may request another state.

4.5.8.1.5 Aggregate states

This subclause defines the possible states of an aggregate.

- a) *Aggregate*. The state in which a unit is broadcasting an Aggregate State PDU only for itself. Aggregates in this state shall not broadcast PDUs for any sub-parts of the aggregate. Aggregate units are usually simulated at the aggregate-level.
- b) *Fully disaggregate*. The state in which a unit is broadcasting Entity State PDUs for its entities, but it does not consistently issue Aggregate State PDUs for itself. It is possible that some simulations will not know about the aggregate if they enter the exercise after the aggregate stops issuing Aggregate State PDUs. The advantage of this state is that it allows the simulation application representing the aggregate to transfer control of the constituent entities and ignore the unit until it is requested to aggregate. The simulation application representing the aggregate also does not have to correlate the state information of aggregates and their constituent entities. These entities are capable of normal interaction with other entities at the virtual level.
- c) *Disaggregate*. The state in which a unit is consistently broadcasting Entity State PDUs for all of its entities and Aggregate State PDUs for itself. This state requires more of the simulation application representing the aggregate, because it must maintain the correlation between the aggregate-level and entity-level representations of the aggregate. This state also keeps other simulations updated as to the state of the aggregate. Entities in a disaggregate unit are capable of normal interaction with other entities at the virtual level.
- d) *Pseudo-disaggregate*. The state in which a unit is broadcasting Entity State PDUs for its entities and is broadcasting Aggregate State PDUs for itself, but the entities in the aggregate are not controlled at the virtual level. These entities are not capable of full interaction with other entities. They may be seen and targeted, but they may not respond normally to events in the virtual world. This state of disaggregation reduces the processing power needed to visually represent aggregate units at the virtual level.
- e) *Partially disaggregate*. The state in which a unit is between aggregate and disaggregate. A partially disaggregate unit broadcasts PDUs for some parts of the aggregate, but it is not broadcasting Entity State PDUs for all of its entities. A unit having one or more subaggregates that are issuing Aggregate State PDUs is in the partially disaggregate state. A unit having some entities that are issuing Entity State PDUs and other entities that are silent is also in the partially disaggregate state. The unit shall broadcast its Aggregate State PDU to represent the multiple parts of the aggregate. This state

requires the simulation application representing the aggregate to correlate the multiple representations of the unit. (See 4.5.8.1.8 for correlation details.)

4.5.8.1.6 Aggregate hierarchy

The aggregation capability of the Entity Management protocol is designed to indicate hierarchical grouping of entities. An aggregate may consist of subaggregates and/or entities. Aggregates should be used to indicate a common characteristic among the entities and sub-aggregates. Entities and aggregates may be grouped together for a number of reasons. The most common method of grouping entities is by representing their military hierarchy. For example, one Aggregate State PDU could represent one tank battalion that consists of three tank companies. These three tank companies could be represented by three Aggregate State PDUs with 20 tanks in each.

4.5.8.1.7 Dead reckoning

A simulation that wishes to track an aggregate unit shall use a first-order dead reckoning algorithm [IEEE Std 1278.1-1995 Annex B algorithm DRM (FPW) or DRM (FPB)].

4.5.8.1.8 Correlation issues

When a unit is in the disaggregate or partially disaggregate state, it is represented at both the aggregate level and the entity level. These two representations need to be correlated in some way. The Entity Management protocol does not specify a specific method to correlate these representations, but it does require that a method of correlation be determined before the exercise starts. The following serves as a guide to users of the Entity Management protocol in determining a correlation method.

4.5.8.1.8.1 Fields to correlate

The following is a list of the fields in the Aggregate State PDU that might change due to events at the virtual level. These fields may or may not be correlated with the virtual world depending on the pre-exercise correlation specifications.

- a) *Position*. As the positions of the entities change, the position of the aggregate should also change.
- b) *Velocity*. As the velocities of the entities change, the velocity of the aggregate should also change.
- c) *Orientation*. As the orientations of the entities change, the orientation of the aggregate should also change.
- d) *Boundaries*. As the positions of the entities change, the boundaries of the aggregate should also change.
- e) *Entity ID List*. As entities are destroyed or disabled, they may or may not be removed from the aggregate.

4.5.8.1.8.2 Timing to correlate

The exercise participants must determine at what time interval the Aggregate State PDU should be transmitted. The participants must also determine the accuracy of the calculations. For example, the simulation application representing the aggregate could calculate the current position of the aggregate from the latest Entity State PDUs, or it could use Entity State PDUs that were broadcast 30 seconds previously.

4.5.8.1.8.3 Entities to correlate

The exercise participants must determine which entities will be included in the calculations of the new aggregate state information. For example, all the entities in the Entity ID List could be included in the calculations, or the calculation could include only those entities that are still active. The exercise participants shall often determine whether destroyed or disabled entities will be left in the Entity ID List.

4.5.8.1.8.4 Example of a correlation method

This example is provided for two reasons: to show the issues that need to be determined by the exercise participants and to provide a method that can be used if the exercise participants cannot agree on another method.

- a) Fields to correlate
 - 1) Position
 - 2) Velocity
 - 3) Orientation
 - 4) Boundaries
 - 5) Entity ID List
- b) Timing
 - 1) When a unit is in the disaggregate state, the Aggregate State PDU should be transmitted at least once every AGG_HRT_BEAT_TIMER seconds.
 - 2) When a unit is in the aggregate state, the Aggregate State PDU should also be transmitted at least once every AGG_HRT_BEAT_TIMER seconds.
 - 3) When a unit is in the disaggregate state, the fields in the Aggregate State PDU should be calculated using entity state information that is less than 2 x HRT_BEAT_TIMER seconds old.
- c) Entities to correlate
 - 1) Any entity that is destroyed or disabled should be removed from the Entity ID List and shall not be included in the aggregate state information.
 - 2) All entities referenced in the Entity ID List should be included in the calculations of the aggregate state information.

4.5.8.1.9 Changing the state of an aggregate (simulation initiated)

The PDUs that shall be sent when a simulation wants a unit to change state are as follows:

- a) *Action Request PDU*. The simulation desiring a change of state shall transmit an Action Request PDU.
- b) *Action Response PDU*. The aggregate shall respond with an Action Response PDU. If the aggregate is uncertain that it should complete the request, then it shall respond with a pending status.
- c) *Event Report PDU*. If the exercise participants specified the requested state as one to be reported, then the aggregate shall issue an Event Report PDU. If the action is pending, then the aggregate shall wait AGG_RESPONSE_DFLT seconds for other simulations to request the current state.
- d) *Aggregate State PDU*. If the state change is completed, then the aggregate shall issue an Aggregate State PDU to indicate the new state of the unit.

4.5.8.1.10 Changing the state of an aggregate (aggregate initiated)

The PDUs that shall be sent when the aggregate wants to change state are as follows:

- a) *Event Report PDU*. If the exercise participants specified the new state as one to be reported, then the aggregate shall issue an Event Report PDU. If the action is pending, then the aggregate shall wait AGG_RESPONSE_DFLT seconds for other simulations to request the current state.
- b) *Aggregate State PDU*. If the state change is completed, then the aggregate shall issue an Aggregate State PDU to indicate the new state of the unit.

4.5.8.2 IsGroupOf PDU

The IsGroupOf PDU shall communicate information about the individual states of a group of entities, including state information that is necessary for the receiving simulation applications to represent the issuing group of entities in the simulation applications' own simulation.

4.5.8.2.1 Issuers of the IsGroupOf PDU

The IsGroupOf PDU may be issued by

- a) A node server for the purpose of communications bandwidth reduction
 - 1) The node server shall receive Entity State PDUs from simulation applications communicating on the local network and generate a single IsGroupOf PDU for each appropriate or logical grouping of entities.
 - 2) An appropriate or logical grouping of entities may be all entities of a specific category within the local network, entities of a specific category separated by force identification, or entities of a specific category grouped within identified tactical units.
- b) A computer-generated forces (CGF) application of either automated or semiautomated force type

4.5.8.2.2 Information contained in the IsGroupOf PDU

The IsGroupOf PDU shall contain the following information:

- a) Identification of the group of entities represented by the PDU
- b) Category of the entity types represented by the PDU
- c) Number of entities grouped within the PDU
- d) Group reference point
- e) Grouped entity descriptions consisting of entity state information, formatted as a function of the category of the grouped entity, including
 - 1) Individual entity identification
 - 2) Individual entity location described in offset from the group reference point
 - 3) Entity appearance
 - 4) Entity orientation
 - 5) Entity speed
 - 6) Entity weapons or articulation parameters appropriate to the entity category
 - 7) (Optional) Logistics status indicators of fuel, ammunition, and maintenance

4.5.8.2.3 Issuance of the IsGroupOf PDU

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

- a) The discrepancy between the actual state of any grouped entity (as determined by its own internal model) and its dead reckoned state (state using specified dead reckoning algorithms) exceeds a predetermined threshold. This threshold includes changes in position/orientation and articulation parameter information.
- b) Change in the grouped entity's appearance occurs.
- c) A change in the number or identification of the grouped entities occurs.
- d) A predetermined length of real-world time has elapsed since the issuing of the last IsGroupOf PDU. This value and associated tolerance may be established at the start of an exercise or during the exercise by means of Simulation Management PDUs. If no value is established, the default values shall be HRT_BEAT_TIMER. A default tolerance of $\pm 10\%$ shall apply to this value.

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

4.5.8.2.3.1 Issuance of a new IsGroupOf PDU by a node server

When the IsGroupOf PDU is generated by a node server, the new IsGroupOf PDU shall incorporate the best current actual state information possessed by the node server and may include a polling of actual state information from the grouped entities.

4.5.8.2.3.2 Issuance of a new IsGroupOf PDU by a CGF application

When the IsGroupOf PDU is generated by a CGF application, the new IsGroupOf PDU shall incorporate the current actual state information of all entities extant in the grouping.

4.5.8.2.4 Receipt of the IsGroupOf PDU

Actions upon receipt of the IsGroupOf PDU are determined by the character of the receiving entity.

4.5.8.2.4.1 Receipt of the IsGroupOf PDU by a simulation application

Upon receipt of the IsGroupOf PDU, a simulation application shall determine, for each of the entities grouped within the IsGroupOf PDU, whether the PDU contains more current information than is currently being used to model those entities within the group being transmitted. If so, the simulation application shall use the information contained therein to model the position, orientation, and appearance of the entities contained within the IsGroupOf PDU. Otherwise, the PDU shall be discarded.

4.5.8.2.4.2 Receipt of the IsGroupOf PDU by a node server

Upon receipt of the IsGroupOf PDU, a node server shall determine, for each of the entities grouped within the IsGroupOf PDU, whether the PDU contains more current information than that currently being used to model those entities within the group being transmitted. If so, for each entity where the information is more current, the communications application shall generate, on the local network, an Entity State Update PDU (see 4.5.2.4). If the newly received entity data do not exceed the parameters of the dead reckoned state of the entity, the communications application shall discard that state information, unless the Simulation Manager (SM) requires that all entity states shall be updated whenever any IsGroupOf PDU is received. If so, an Entity State Update PDU shall be issued. When all appropriate Entity State Update PDUs have been issued to the local network, the IsGroupOf PDU shall be discarded.

4.5.8.2.5 Dead reckoning

A first-order dead reckoning algorithm [IEEE Std 1278.1-1995 Annex B algorithm DRM (FPW) or DRM (FPB)] shall be employed to estimate the position and orientation of the group in order to limit the number of PDUs to be issued.

4.5.8.3 Entity handover

The Transfer Control Request PDU in conjunction with three Simulation Management with Reliability PDUs provides a capability for the transfer of control of entities from one simulation application to another in a DIS exercise. An entity may be a traditional entity, a process, a cultural feature, or any other object within a DIS exercise for which control may pass from one simulation application to another for any reason.

The transfer of control shall be accomplished through a series of request and response messages between two simulation applications. The request for handover shall be initiated by the application controlling the entity or by an application desiring control of an entity that is controlled by another application. One application may initiate the mutual exchange, or swapping, of control of the two entities. The identification of the entity shall be maintained throughout the existence of the entity; it shall not change as a result of the transfer. See 4.5.8.3.4 for illustrations of these types of exchanges.

The three Simulation Management with Reliability PDUs (see 4.5.11) used with the Transfer Control Request PDU are

- a) Set Record-R PDU
- b) Record-R PDU
- c) Acknowledge-R PDU

When these Simulation Management with Reliability PDUs are used in the Entity Management protocol, they shall specify the Entity Management protocol in the Protocol Family field of the PDU header.

Reliability service levels (see 4.5.11.2) are available for use in handing over control of an entity utilizing the entity handover capability of the Entity Management protocol. The level of reliability service to be used in the exchange of PDUs required to hand over an entity shall be specified in the Transfer Control Request PDU. Use of the reliability service is highly recommended for all mutual exchanges of control.

4.5.8.3.1 Transfer Control Request PDU

The Transfer Control Request PDU shall be used to communicate a request for control of an entity belonging to another simulation application.

4.5.8.3.1.1 Information contained in the Transfer Control Request PDU

The Transfer Control Request PDU shall contain the following information:

- a) Standard simulation management header
- b) Request identification for the transfer of control being requested
- c) Level of reliability required for this transaction
- d) Type of transfer being requested
- e) Identification of the entity for which transfer of control is being requested
- f) Number of records, the identification of the records, and the particular parameter values

4.5.8.3.1.2 Issuance of the Transfer Control Request PDU

The Transfer Control Request PDU shall be issued by an application desiring control of another application's entity or by an application wishing to relinquish control of an entity to another application. An application wishing to relinquish control of an entity may issue multiple Transfer Control Request PDUs in an attempt to find an application willing to take control of the entity. Additional data required to control the entity and perhaps only available to the controlling application shall be included in the data portion of the Transfer Control Request PDU. Examples of such data include fuel levels, internal stores, oil pressure, etc.

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

4.5.8.3.1.3 Receipt of the Transfer Control Request PDU

Upon receipt of the Transfer Control Request PDU, an Acknowledge-R PDU shall be returned to indicate the ability to comply with the transfer of control of the entity. If the receiving application does not intend to relinquish control, the receiving application shall also issue an Acknowledge-R PDU. Upon receipt of the Acknowledge-R PDU approving the transfer, the receiving application shall initiate simulation of the entity. The receiving application may not be able to comply with all the parameter values specified within the record portion of the Transfer Control Request PDU; however, the application must make the best effort possible to match the specified conditions. The application taking control shall issue a Record-R PDU to confirm the parameter values used.

If the Transfer Control Request PDU indicates a mutual exchange, or swapping, of control of two entities, the application receiving the Transfer Control Request PDU shall reply with a Set Record-R PDU describing the entity it will transfer and then shall issue an Acknowledge-R PDU. Both applications shall issue a Record-R PDU to confirm the data used.

The Acknowledge-R PDU shall be issued using a best effort multicast communication service.

4.5.8.3.2 Set Record-R PDU

The Set Record-R PDU shall be used in the entity handover capability of the Entity Management protocol to exchange data between the simulation application requesting to transfer control of an entity to itself and the current simulation application owning the entity.

4.5.8.3.2.1 Information contained in the Set Record-R PDU

The Set Record-R PDU shall contain the following information:

- a) Standard simulation management header
- b) Request identification number
- c) Number of records, the identification records, and the specific parameter values
- d) Level of reliability service to be used

4.5.8.3.2.2 Issuance of the Set Record-R PDU

A Set Record-R PDU is used in the entity handover sequence only when the Transfer Control Request PDU indicates a mutual exchange, or swapping, of control of two entities. The Request ID field of the Set Record-R PDU shall contain the same value as in the Transfer Control Request PDU that initiated the handover of this entity. The first data record shall contain the ID of the application assuming control of the entity. Additional data required to control the entity and perhaps otherwise only available to the controlling application shall be included in the Set Record-R PDU.

The Set Record-R PDU shall be issued using a best effort multicast communication service.

4.5.8.3.2.3 Receipt of the Set Record-R PDU

Upon receipt of the Set Record-R PDU, the receiving application shall initiate simulation of the entity. The application may not be able to comply with all the parameter values specified within the Set Record-R PDU; however, the application must make the best effort possible to match the specified conditions.

4.5.8.3.3 Record-R PDU

The Record-R PDU shall be used in the entity handover capability of the Entity Management protocol in response to a Transfer Control Request PDU or a Set Record-R PDU.

4.5.8.3.3.1 Information contained in the Record-R PDU

The Record-R PDU shall contain the following information:

- a) Standard simulation management header
- b) Request identification number
- c) Number of records, the identification of the records, and the specific parameter values
- d) Level of reliability service to be used

4.5.8.3.3.2 Issuance of the Record-R PDU

The Record-R PDU, when used in the Entity Management protocol, shall be issued by an application upon receipt of a Transfer Control Request PDU containing parameter values in the record portion of the PDU or upon the receipt of a Set Record-R PDU. The Record-R PDU shall be issued to indicate receipt of the data

and to confirm the values used for each of the specified data fields in the Transfer Control Request PDU or in the Set Record-R PDU.

The Record-R PDU shall be issued using a best effort multicast communication service.

4.5.8.3.3 Receipt of the Record-R PDU

Upon receipt of the Record-R PDU, the controlling application shall continue to issue the updates (i.e., Entity State PDU, Environmental Process PDU) and all related PDU traffic until an update is received from the application acquiring control of the entity. Once an update is received from the acquiring application, the application formerly controlling the entity shall remove the entity from its application and shall cease all PDU transmissions for the entity. The Record-R PDU shall serve to verify the transfer will take place without indicating the completion of the transfer. The receipt of the update from the application acquiring control shall indicate completion of the transfer. Having the controlling application continue to issue updates is vital for highly dynamic entities that require frequent updates.

4.5.8.3.4 Exchange illustrations

The PDUs exchanged for the transfer of an entity are illustrated in 4.5.8.3.4.1 through 4.5.8.3.4.3 for the three forms of transfer supported by the Entity Management protocol.

4.5.8.3.4.1 Transfer initiated by controlling application

Figure 22 illustrates the exchange of PDUs utilized to transfer the control of an entity when the application currently controlling the entity initiates the transfer request.

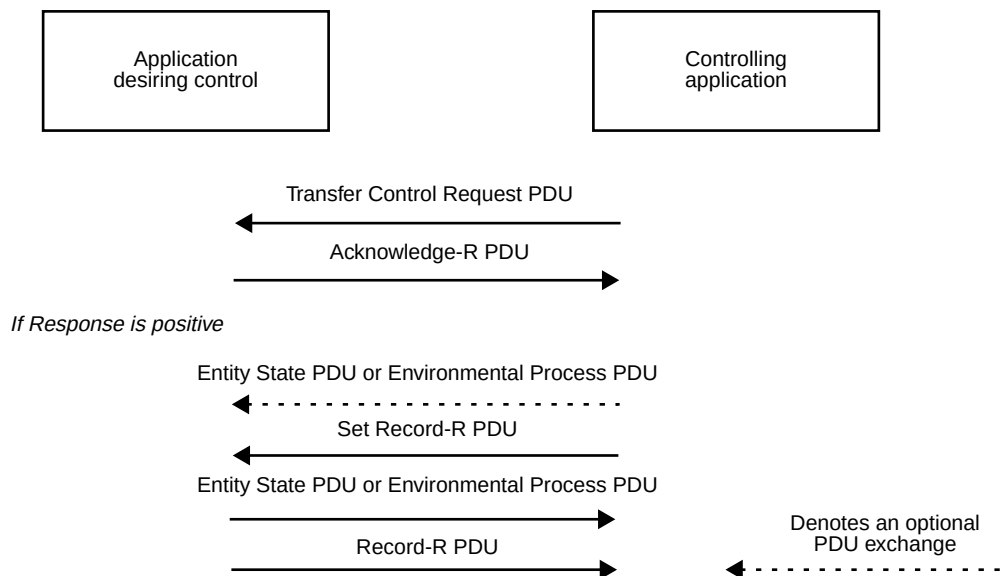


Figure 22—Transfer initiated by controlling application

4.5.8.3.4.2 Transfer initiated by application desiring control

Figure 23 illustrates the exchange of PDUs utilized to transfer the control of an entity when the application desiring control of the entity initiates the transfer request.

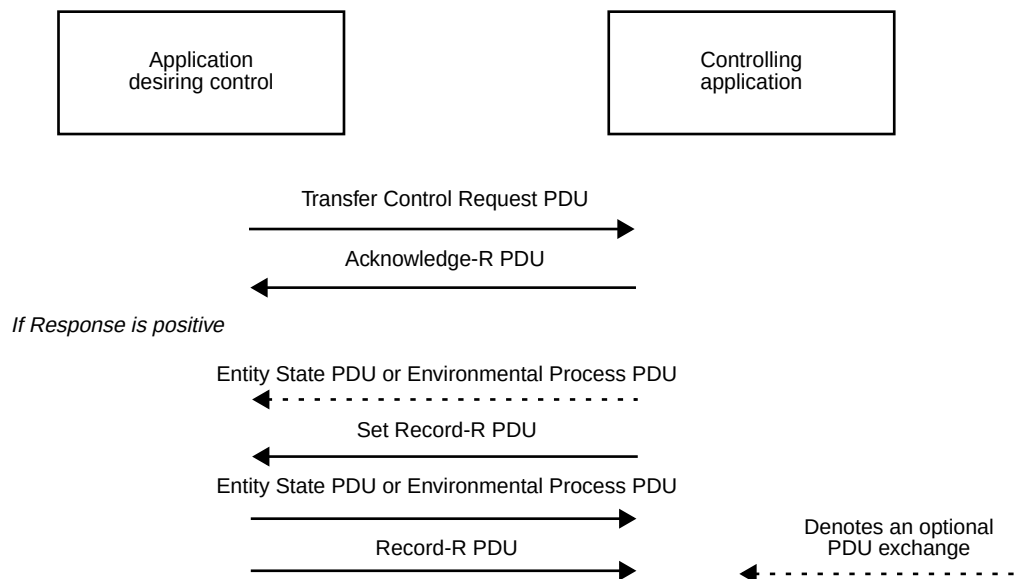


Figure 23—Transfer initiated by application desiring control

4.5.8.3.4.3 Mutual exchange of control transfer

Figure 24 illustrates the exchange of PDUs utilized to transfer the control of an entity or environmental process when there is a mutual exchange between the application initiating the mutual exchange and the application participating in the mutual exchange.

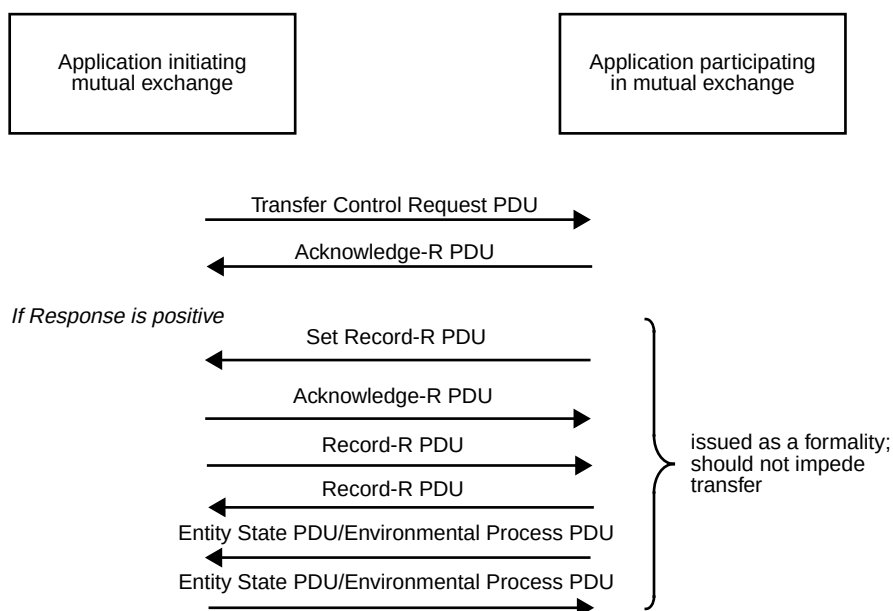


Figure 24—Mutual exchange of control transfer

4.5.8.4 IsPartOf PDU

The IsPartOf PDU shall be used to request hierarchical linkage of separately hosted simulation entities where the originating simulation application requests that the receiving simulation application honor a respective host/part relationship. An entity that becomes a part of another entity is referred to as a part entity from the time it joins the host entity until it leaves the host entity. When it leaves the host entity, it is again referred to as an entity. The originating entity is also referred to as the host entity.

4.5.8.4.1 Information contained in the IsPartOf PDU

The IsPartOf PDU shall contain the following information:

- a) Standard simulation management header where
 - 1) Originating Entity is the entity (host entity) that requests the Receiving Entity to become a part of the host entity.
 - 2) Receiving Entity is the entity requested to become a part of the Originating Entity.
- b) Relationship between the host entity and part entity
- c) Location of the part in the host entity's coordinate system
- d) Identification of the part's station with respect to the host entity
- e) Attached part's entity type

4.5.8.4.2 Issuance of the IsPartOf PDU

The IsPartOf PDU shall be issued by a simulation application to request that an entity from a remotely hosted simulation application be treated as a constituent part of the Originating Entity.

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

4.5.8.4.3 Receipt of the IsPartOf PDU

Upon receipt of an IsPartOf PDU, the receiving simulation application shall make a determination as to the feasibility of honoring the IsPartOf request and shall issue an Acknowledge PDU.

If the requested simulation application is able to comply with the request made in the IsPartOf PDU, the receiving simulation application shall cease issuing Entity State PDUs for its entity (which is now a part of the Originating Entity). The simulation application issuing the IsPartOf PDU shall be responsible for updating the location and orientation information of the part until the part separates from the host entity through deployment, debarkation, or firing events.

Engagement space destruction of the host entity shall imply similar destruction of all of its constituent part entities.

4.5.8.4.3.1 Munition parts

Upon firing, a part entity that is a munition shall no longer be considered a constituent part of the firing platform. See 4.5.3.1 item b) for the requirements addressing tracked and nontracked fired munitions.

4.5.8.4.4 Acknowledgment of the IsPartOf PDU

The IsPartOf PDU receiving application shall acknowledge receipt of the IsPartOf PDU by transmitting an Acknowledge PDU (see 4.5.5.4.5). The Originating Entity ID and Receiving Entity ID fields in the Acknowledge PDU shall be the exact reverse of the corresponding fields in the IsPartOf PDU. When used in the IsPartOf capability of the Entity Management protocol, the Acknowledge PDU shall specify the Entity Management protocol in the Protocol field of the PDU header.

The Request ID field of the Acknowledge PDU sent in reply to an IsPartOf PDU shall contain the value zero.

4.5.9 Minefield

The protocol for communicating information associated with the location, appearance, and other pertinent details of mines and minefields is described in 4.5.9.1 through 4.5.9.4. The Minefield protocol is composed of the following four PDUs:

- a) Minefield State PDU
- b) Minefield Query PDU
- c) Minefield Data PDU
- d) Minefield Response NACK PDU

These PDUs can be used in two protocol modes. One mode called the Minefield Query Response Protocol (QRP) mode uses all four PDUs. The second mode, heartbeat mode, uses only the Minefield State PDU and the Minefield Data PDU. The protocol mode is chosen by the minefield simulation. This choice should be determined prior to the start of an exercise.

The Minefield State PDU provides information defining an aggregate minefield object that consists of a set of individual mines. A Minefield Data PDU is utilized to describe deaggregated mines contained within a minefield.

The QRP mode provides an alternative to the heartbeat-based transmission of the Minefield Data PDU. If the QRP mode is selected, the Minefield Data PDU shall only be transmitted in response to a Minefield Query PDU. The Minefield Query PDU is sent to request information about deaggregated mines within a region. The Minefield Response NACK PDU is used to request retransmission of single Minefield Data PDUs that are not received by the querying simulation. QRP mode (i.e., the use of the Minefield Query and Minefield Response NACK PDUs along with the Minefield State and Minefield Data PDUs) is recommended for DIS exercises with large quantities of mines.

4.5.9.1 Minefield State PDU

The Minefield State PDU shall communicate information about the minefield, including the location, perimeter, and types of mines contained within it.

4.5.9.1.1 Information contained in the Minefield State PDU

The Minefield State PDU shall contain the following information:

- a) Identification of the minefield issuing the PDU
- b) A sequentially incrementing number, starting with 1, to indicate a change in any of the information in items c) through l) below or in the state of any of the mines contained within the minefield
- c) Identification of the force to which the minefield belongs
- d) Number of perimeter points defining the extent of the minefield
- e) Minefield type
- f) Number of types of mines in the minefield
- g) Location of the minefield in the simulated world
- h) Orientation of the minefield in the simulated world
- i) Appearance information for displaying the symbology of the minefield as a doctrinal minefield graphic
- j) Protocol mode that the minefield simulation is using to communicate the minefield data
- k) Coordinates of the perimeter points
- l) Identification of the types of mines in the minefield

4.5.9.1.2 Issuance of the Minefield State PDU

A minefield simulation shall issue a Minefield State PDU when any of the following occur:

- a) A change has occurred in the force ID, location, orientation, perimeter, mine types, or appearance of the minefield.
- b) A change has occurred in the state of a mine in the minefield and a Minefield State PDU has not been sent since the last Minefield Query PDU for this minefield was answered by the minefield simulation. If a Minefield State PDU has been issued since the last Minefield Query PDU was satisfied, a Minefield State PDU shall not be sent when the state of a mine in a minefield changes [unless the state change causes a corresponding change in the minefield covered under item a) above].
- c) A predetermined length of real-world time has elapsed since the issuing of the last Minefield State PDU. This value and associated tolerance may be established at the start of an exercise or during the exercise by means of Simulation Management PDUs. If no value is established, the default value shall be MINEFIELD_DFLT. A default tolerance of $\pm 10\%$ shall apply to this value.

The Minefield State PDU shall be issued using a best effort multicast communication service.

4.5.9.1.3 Receipt of the Minefield State PDU

Upon receipt of a Minefield State PDU, the receiving simulation application operating in QRP mode shall examine the sequence number to determine whether any of the information regarding the minefield has changed. If the sequence number has incremented since receipt of the previous Minefield State PDU, then the new information about the minefield shall be used by the simulation. Otherwise, the PDU shall be discarded. Upon receipt of a Minefield State PDU, the receiving simulation application operating in heartbeat mode shall use the new information about the minefield.

If the appearance information of the minefield indicates the minefield is in the Deactivated state or a predetermined length of time has elapsed since the receipt of the last Minefield State PDU, then all simulations shall remove the minefield from the exercise. This length of time shall be established at exercise start, although it may be changed during the exercise. If no value for this time interval is established, then the default value for this time shall be the value of the Minefield State PDU issuance interval [4.5.9.1.2 item c)] multiplied by HRT_BEAT_MPLIER.

4.5.9.2 Minefield Query PDU

The Minefield Query PDU shall contain information about the requesting entity and the region and mine types of interest to the requesting entity. The Minefield Query PDU shall only be used if the QRP mode is selected.

4.5.9.2.1 Information contained in the Minefield Query PDU

The Minefield Query PDU shall contain the following information:

- a) Identification of the minefield
- b) Identification of the requesting entity
- c) Request identification of the query
- d) Number of perimeter points for the requested area
- e) Number of requesting sensors
- f) Data filter
- g) Request mine type filter
- h) Coordinates of the perimeter points of the requested area
- i) Sensor types

4.5.9.2.2 Issuance of the Minefield Query PDU

The Minefield Query PDU shall be issued to request additional mine information from the minefield simulation when the following conditions exist:

- a) Use of the Minefield Query PDU (i.e., QRP mode) is selected.
- b) Based on a Minefield State PDU, an object (mine type) of interest exists.
- c) A minefield is within or near the sensor's range of the vehicle.

The Minefield Query PDU shall be issued using a best effort multicast communication service.

4.5.9.2.3 Receipt of the Minefield Query PDU

Upon receipt of a Minefield Query PDU, a minefield simulation shall utilize the information in the Minefield Query PDU to provide a tailored response with only the requested information provided. In particular, the minefield simulation shall use the following criteria to determine which mine(s) are sent in a Minefield Data PDU:

- a) Is the mine of the requested mine type?
- b) Is the mine within the perimeter of the requested area?

A Minefield Data PDU shall always be sent even if no mines match the query. If the resulting Minefield Data PDU exceeds the allowable PDU size limit (see IEEE Std 1278.2-1995), the mine information shall be distributed in multiple Minefield Data PDUs.

4.5.9.3 Minefield Data PDU

Information about the location and status of a collection of mines in a minefield is conveyed through the Minefield Data PDU on an individual mine basis.

If the QRP mode is selected, the Minefield Data PDU shall supply only mine data that satisfy conditions of the requesting sensor types and the mine type and geographic filters provided in the Minefield Query PDU. If heartbeat mode is selected, the Minefield Data PDU shall supply mine data for all the mines in the minefield or for mines that have changed (see 4.5.9.3.2.1).

4.5.9.3.1 Information contained in the Minefield Data PDU

The Minefield Data PDU shall contain the following information:

- a) Identification of the minefield
- b) Identification of the requesting entity
- c) Minefield sequence number
- d) Request identification associated with the query being processed
- e) PDU sequence number
- f) Number of PDUs in response set
- g) Number of mines
- h) Number of requesting sensors
- i) Data filter that indicates which of the following optional data (1–13) are present for the mine(s):
 - 1) Ground burial depth offset of the mine
 - 2) Water burial depth offset of the mine
 - 3) Snow burial depth offset of the mine
 - 4) Orientation of the mine
 - 5) Thermal contrast between mine and the surrounding soil
 - 6) Reflectance of the mine
 - 7) Emplacement time of the mine

- 8) Fusing mechanism for the mine
- 9) Scalar Detection Coefficient for the sensor types
- 10) Paint scheme of the mine
- 11) Number of trip/detonation wires of the mine
- 12) Number of vertices of the trip/detonation wires
- 13) Vertices of the trip/detonation wires
- j) Type of mine contained in the PDU
- k) Requesting sensor type(s)
- l) Location of the mine(s)
- m) Mine entity identifier

4.5.9.3.2 Issuance of the Minefield Data PDU

The Minefield Data PDU shall be issued using a best effort multicast communication service.

The conditions for issuance of the Minefield Data PDU vary depending upon the selection of the protocol mode.

4.5.9.3.2.1 Heartbeat mode issuance criteria

The minefield simulation shall issue a Minefield Data PDU under the following conditions when the heartbeat mode is selected:

- a) A change has occurred in the minefield ID associated with a mine; or
- b) A change has occurred in the state of a mine; or
- c) A predetermined length of real-world time has elapsed since the issuing of the last Minefield Data PDU describing a specific mine. This value and associated tolerance may be established at the start of an exercise or during the exercise by means of Simulation Management PDUs. If no value is established, the default value shall be MINEFIELD_DATA_DFLT. A default tolerance of $\pm 10\%$ shall apply to this value.

The sending of changes [item a) or b) above] is controlled by MINEFIELD_CHANGE_DFLT. This value shall be between 0 and MINEFIELD_DATA_DFLT. If the value is 0, changes are sent immediately when they occur. If the value is MINEFIELD_DATA_DFLT, change notification is only accomplished via the heartbeat. If the value is between 0 and MINEFIELD_DATA_DFLT, notification of multiple changes shall be sent together between heartbeats at the indicated rate if any changes existed to be sent. MINEFIELD_CHANGE_DFLT represents a predetermined length of real-world time that has elapsed since the last Minefield Data PDU would have been sent for a change notification if any changes existed to be sent. This value and associated tolerance may be established at the start of an exercise or during the exercise by means of Simulation Management PDUs. If no value is established, the default value shall be MINEFIELD_DATA_DFLT (i.e., data are sent only on heartbeat and not when changes occur).

A Minefield Data PDU shall contain mine information for only a single type of mine. A Minefield Data PDU may contain data for a single mine or multiple mines. The minefield simulation should make a best effort to combine data for multiple mines into a single Minefield Data PDU in order to reduce network loading. The PDU sequence number is not used when operating in the heartbeat mode.

4.5.9.3.2.2 QRP mode issuance criteria

A minefield simulation shall issue a Minefield Data PDU when operating in the QRP mode only in response to the receipt of a Minefield Query PDU.

A Minefield Data PDU shall contain mine information for a single type of mine. Multiple Minefield Data PDUs shall be issued if multiple mine types match a query or if the mine information for a single type of mine does not fit in a single PDU. The set of Minefield Data PDUs necessary to contain all of the requested mine information for the requested mine type and within the requested perimeter shall be sent in response to a Minefield Query PDU. The Number of PDUs field shall indicate how many Minefield Data PDUs are being sent in the response set of Minefield Data PDUs. The PDU sequence number shall specify an individual Minefield Data PDU within this set.

In QRP mode, the minefield simulation must be able to retransmit individual Minefield Data PDUs in response to a Minefield Response NACK PDU.

4.5.9.3.3 Receipt of the Minefield Data PDU

4.5.9.3.3.1 Receipt of the Minefield Data PDU in heartbeat mode

Upon receipt of a Minefield Data PDU, the receiving simulation application operating in heartbeat mode shall use the new information about the minefield.

If a predetermined length of time has elapsed since the last Minefield Data PDU representing the given mine was received, the simulation application shall remove the model of that mine from its simulation. This value and associated tolerance may be established at the start of an exercise or during the exercise by means of Simulation Management PDUs. If no value is established, the default value shall be `MINEFIELD_DATA_TIMEOUT_DFLT`. A default tolerance of $\pm 10\%$ shall apply to this value.

4.5.9.3.3.2 Receipt of the Minefield Data PDU in QRP Mode

Upon receipt of a Minefield Data PDU, the receiving simulation application operating in QRP mode shall examine the minefield sequence number to determine whether any of the information about the mines in the minefield has changed (the sequence number could be incremented for reasons other than changes in individual mines). If the sequence number has incremented since receipt of the previous Minefield Data PDU, then the new information about the minefield shall be used by the simulation. Otherwise, the PDU shall be discarded.

When operating in QRP mode, the receiving simulation application shall remove the model of a mine if the minefield referenced by the Minefield ID is deleted. The simulation application shall internally represent a mine as obsolete if a Minefield Data PDU with the same Minefield ID, the same Mine Type, and a newer sequence number is received that does not contain the mine. This step is due to the fact that a simulation cannot be sure of the existence and state (including location) of a particular mine that it previously acquired, since the Minefield Data PDU shall contain only data for the specified query region. If a Minefield Data PDU is subsequently received that provides current data for the obsolete mine, the old state information shall be replaced with the new state information and the state of the mine updated from obsolete to current. The simulation application shall not utilize obsolete mine data but may maintain previously generated internal state information for obsolete mines indefinitely. A possible use of internal state information is to provide consistent detection probabilities across acquisitions should the mine be updated to current.

4.5.9.4 Minefield Response NACK PDU

The Minefield Response NACK PDU shall contain information about the requesting entity and the PDU(s) that were not received in response to a query. The Minefield Response NACK PDU shall be used only if the QRP mode is selected.

4.5.9.4.1 Information contained in the Minefield Response NACK PDU

The Minefield Response NACK PDU shall contain the following information:

- a) Identification of the minefield
- b) Identification of the requesting entity
- c) Request identification of the corresponding query
- d) Number of missing PDU(s)
- e) Missing PDU sequence number(s)

4.5.9.4.2 Issuance of the Minefield Response NACK PDU

The Minefield Response NACK PDU shall be issued to request Minefield Data PDU(s) that were not received by the requesting simulation in response to a Minefield Query PDU. The Minefield Response NACK PDU shall be issued after a predetermined length of real-world time has elapsed since the receipt of a Minefield Data PDU in response to a query if the number of PDU(s) indicated have not all been received. This time value and associated tolerance may be established at the start of an exercise or during the exercise by means of Simulation Management PDUs. If no value is established, the default value shall be MINEFIELD_RESPONSE_DFLT. A default tolerance of $\pm 10\%$ shall apply to this value.

The Minefield Response NACK PDU shall be issued using a best effort multicast communication service.

4.5.9.4.3 Receipt of the Minefield Response NACK PDU

Upon receipt of a Minefield Response NACK PDU, a minefield simulation operating in QRP mode shall utilize the information in the Minefield Response NACK PDU to retransmit the requested PDU(s).

4.5.10 Synthetic environment

The Synthetic Environment protocol supports the simulation of environmental conditions that change during the life of a distributed simulation exercise. This protocol contains the following PDUs:

- a) *Environmental Process PDU*. Communicates simple environment variables, small scale environmental updates, and embedded processes.
- b) *Gridded Data PDU*. Transmits information about large-scale or high-fidelity spatially and temporally varying ambient fields and about environmental processes and features.
- c) *Point Object State PDU*. Transmits specific subset of information about environment objects described by a point.
- d) *Linear Object State PDU*. Transmits specific subset of information about environment objects described by lines.
- e) *Areal Object State PDU*. Transmits specific subset of information about environment objects described by areas.

The Synthetic Environment protocol provides two approaches for the simulation of environmental conditions. One approach is a process approach and the second is an object approach. These approaches and the associated protocols are described in 4.5.10.1 and 4.5.10.2, respectively.

4.5.10.1 Process approach to synthetic environment simulations

The following rules shall be followed when utilizing the process approach to the Synthetic Environment protocol in DIS:

- a) The application responsible for simulating a particular environmental condition or conditions shall output Environmental Process and Gridded Data PDUs to represent the current state and geometry of the environmental condition or conditions.
- b) Receiving applications that are affected by the environmental condition(s) shall use the parametric physical data in the Environmental Process and Gridded Data PDUs to drive the appropriate effects models within their simulations.