Technical Document 1801
February 1990

Distributed Operating System Experiment (DOSE) Application User's Manual

Manchi J. Gadbois

NAVAL OCEAN SYSTEMS CENTER

San Diego, California 92152-5000

J. D. FONTANA, CAPT, USN
Commander

R. M. HILLYER
Technical Director

ADMINISTRATIVE INFORMATION

The work in this document was performed by the Distributed System Branch of the Naval Ocean Systems Center under Office of Naval Technology Block Funding.

Released by
G. B. Myers, Jr., Head
Distributed Systems Branch

Under authority of
A. G. Justice, Head
Information Processing
and Displaying Division

PK

CONTENTS

FIGURES

1.0 INTRODUCTION

The Naval Ocean Systems Center (NOSC) initiated Cronus research with Release 1.0 in 1987 (Sullivan and Anderson, 1988). Cronus, a distributed computing environment designed specifically for a command and control environment, was developed for Rome Air Development Center (RADC) and NOSC by BBN Systems and Technologies Corporation. Cronus operates on a variety of different hardwares and runs at the user level above the host's native operating system (Berets and Sands, 1989). The Distributed Operating System Experiment (DOSE) Version 4.0 application was designed and implemented by NOSC to exercise and observe the facilities provided by Cronus. DOSE application demonstrates the ability to map a Navy application onto a distributed system. This document describes the functionalities and capabilities of each component of the unclassified version of the DOSE application. The document also provides instructions on how to run the application in the Cronus environment (Release 1.3 or newer release) and recommended testbed layouts for a demonstration according to the number of host machines being used.

The DOSE application is composed of four components: **Parser Manager, Track Report Manager, Graphics Map Client,** and **Database Monitor Client.** The scenario used to test and demonstrate the DOSE application is an outer air-battle scenario generated at NOSC. The scenario file is sent to a Network Interface Unit (NIU) client that packages the raw Navy Tactical Data System (NTDS) data read in from the Data Terminal Set (DTS) into blocks. This, in turn, sends LINK 11, a high-speed, high-frequency or ultrahigh-frequency data transmission net that connects ships and aircraft equipped with the NTDS, to the Parser Manager. The primary function of the Parser Manager is to convert these Link 11 binary messages into ASCII text. Due to the sensitive nature of the Link 11 data formats, there exists a classified and an unclassified version of the Parser Manager. In the unclassified version, the operation of parsing blocks of binary data is unimplemented. Instead, previously parsed track information is stored in a file to be read by the Parser Manager. The track information is passed to a Track Report Manager for storage in its object database. Cronus object database is not a relational database system such as Informix or Oracle. Cronus database is a stable storage used by an object manager (Track Report Manager) to maintain the instance variables (member of the common characteristics for a set of objects), replication information, and access control lists that make up the objects that the manager manages. Existing multiple copies of the Track Report Manager may be concurrently running with replicated data objects. For each new track information received, a new object is created by the Track Report Manager. For track updates, the Track Report Manager searches through its object database using the track number as the key and overwrites the appropriate object content with the new information. The track information can then be displayed by the Graphics Map Client using an unclassified NTDS symbology. The Track Report Manager also services the Database Monitor Client that monitors the number of objects in each of the Track Report Managers' replicated object databases. The Database Monitor Client also invokes (broadcasts) Cronus' *RunUpdateDaemon* that updates the replicated objects (for consistency) in the object database of all the Track Report Managers.

There are two companion documents to this User's Manual. The *Distributed Operating System Experiment 1988 Source Code* (Kwong, 1989) provides an overview of the DOSE application and the unclassified source code to the application. The *Distributed Operating System Experiment (DOSE) Application Installation Manual* (Gadbois, 1990) describes the procedures for installing the DOSE application.

This document assumes the user is familiar with the Cronus computing environment and the UNIX environment.

2.0 DOSE COMPONENTS

2.1 PARSER MANAGER

The Parser Manager was designed and developed under UNIX using transmission control protocols/interprocess communication (TCP/IP) and then redesigned to be a Cronus manager. To increase survivability, portability, and availability, the Parser Manager was written with the capability to run on UNIX and VMS machines. The Parser Manager is currently running on Sun Workstations (UNIX) and Micro VAX II (VMS) in the NOSC testbed.

The Parser Manager services an NIU client that sends the Parser manager raw Link 11 data. The primary function of the Parser Manager is to convert these Link 11 binary messages into ASCII text. The ASCII text is sent to the Track Report Manager for storage. The subset of Link 11 messages handled are equivalent to the subset handled by the Dots 4.0 Link 11 decoder implemented on a HP9020 and prototyped on the USS *Carl Vincent*. The messages that the Parser Manager does not handle are ignored. Erroneous data messages are also ignored. These binary messages contain unclassified information about tracks such as latitude, longitude, type, and category. Even though the Parser Manager is a Cronus manager, it does not save or store track information.

Due to the sensitive nature of the Link 11 data formats, an unclassified version of the Parser Manager was created for public distribution. The parsing operation of the binary data is unimplemented. In its place is an operation that reads previously parsed unclassified track information (ASCII characters) stored in a file. The Parser Manager then passes the track information to the Track Report Manager for storage.

The Parser Manager performs the following eight operations:

Setdlrp Due to the nature of the Link 11 data, the user must set the Data Link Reference Point (DLRP) prior to sending data to the Parser Manager. The DLRP is set by invoking this operation via the Cronus support and debug tool, *tropic*. The parameters to the operation are the latitude and longitude (in minutes) of the new DLRP. The Setdlrp operation is unimplemented in the unclassified version of the Parser Manager.

Getdlrp The latitude and longitude information (in minutes) of the DLRP can be obtained by invoking this operation via tropic. The Getdlrp operation is unimplemented in the unclassified version of the Parser Manager.

SetDisplayTextFlag One of the means to debug the Parser Manager and check for correctness of the parsing algorithm is to textually display the parsed track information on the terminal. This capability can be controlled by the user by invoking the SetDisplayTextFlag operation via tropic and setting the flag to either on or off. The flag is set to off by default.

SetSendtrmFlag The Parser Manager passes the parsed track information to the Track Report Manager for storage. The Track Report Manager does not have to be running for debugging and testing purposes of the Parser Manager. However, if the Track Report Manager is not running, the Parser Manager will attempt to invoke it, resulting in repeated timeouts and causing a lock-up. To remedy this situation, the SetSendtrmFlag operation allows the user to set the flag to off, which signals the Parser Manager not to send track information to the Track Report Manager. The operation is invoked via tropic and the flag is set to on by default.

SetGraphicsFlag A Cronus manager is usually a hidden service. For demonstration purposes, the Parser Manager has a graphics capability that displays monitoring information about its status and the

data flowing through it (figure 1). The SetGraphicsFlag operation allows the user to activate the Parser Manager's graphics by setting the flag to on via tropic. The flag is set to off by default.

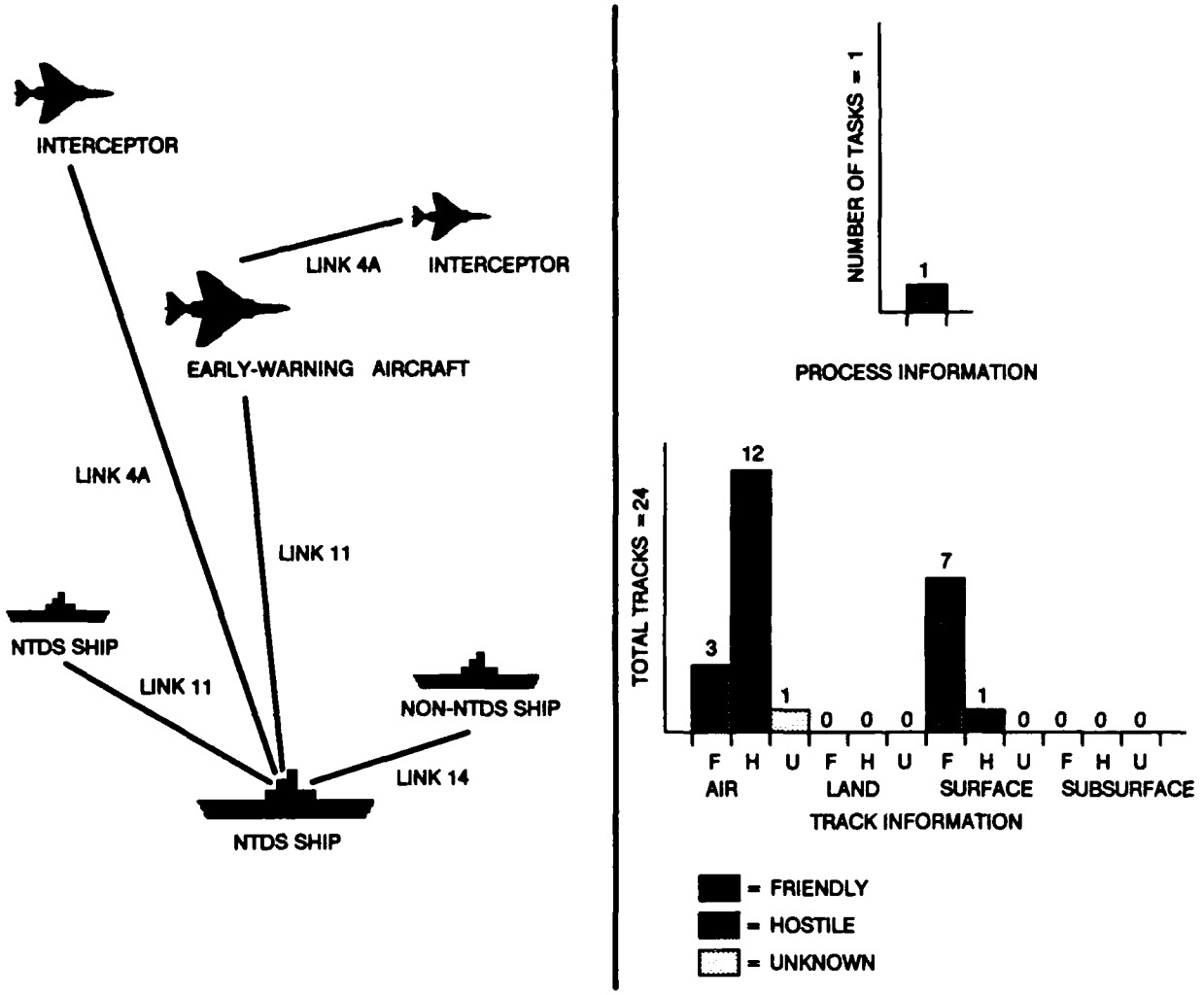

Figure 1. Parser Manager graphical display.

SetDemoFlag The Parser Manager processes many messages at a time and immediately passes the information to the Track Report Manager. Besides serving the Parser Manager, the Track Report Manager also services the Graphics Map Client and the Database Monitor Client. In addition, the Track Report Manager manages replicated data objects when multiple copies of the manager are running. The Track Report Manager also provides graphics display capability during the demonstration. Thus, the many demands made on the Track Report Manager will slow down its processing, which will result in a backlog in attempting to handle the information being sent by the Parser Manager. To avoid timeout errors and overloading the Track Report Manager, the Parser Manager will calculate and set a timeout variable based on a prediction of the time the Track Report Manager will take to process each track information. Thus, the Parser Manager will sleep according to the timeout variable between invocations on the Track Report Manager. The timeout variable defaults to 30 seconds. This capability is activated by setting the flag on to the SetDemoFlag operation via tropic. The flag is set to off by default.

ParseBlock An NIU client running on an access machine, which is an interface from a Link 11 data terminal set to the Cronus environment, invokes the ParseBlock operation sending raw Link 11 data to the Parser Manager. The operation parses the block of Link 11 data contained in the array of Link 11 messages and sends the parsed ASCII data to the Track Report Manager. The parsed data are sent in blocks to the Track Report Manager to reduce the number of invocations. The ParseBlock operation is unimplemented in the unclassified version of the Parser Manager.

ReadTrack Due to the sensitivity of the Link 11 data formats, the ParseBlock operation is replaced by the ReadTrack operation in the unclassified version of the Parser Manager. The ReadTrack operation reads already parsed track information from an ASCII file called *trackfile*. The file groups the track information into blocks. Each block contains a varied number of track information (figure 2). The beginning of each block is a number that indicates the number of tracks in that block. Each line in the file represents the data that defines the track: track number, latitude of the track, longitude of the track, bearing from the DLRP, depth or height of the track, latitude of DLRP, longitude of DLRP, type of platform, identity or category of the track, greenwich mean time, track quality, latitude direction (north or south), longitude direction (east or west), course, speed, range from DLRP, and nuclear status. The Parser Manager reads in the block of track information and partitions the block into an array with a maximum of eight tracks. These eight tracks are sent to the Track Report Manager via invoking the operation, *TrackUpdate*. After the Parser Manager finishes sending all the tracks in the block to the Track Report Manager, the Parser Manager reads in the next block of track information. This operation, which does not perform any parsing action, is invoked via tropic.

2.2 TRACK REPORT MANAGER

Designed specifically for the DOSE application, the Track Report Manager is an object database manager that manages Navy track objects. The Track Report Manager, which runs only on UNIX machines, is a fully replicated Cronus manager that services the Parser Manager, Graphics Display Client, and Database Monitor Client. The Track Report Manager can be ported to run on VMS machines like the Parser Manager. However, the porting has not been done because of the desire for Track Report Manager's graphics capability.

The Track Report Manager manages a database of data objects that are replicated with each copy of the Track Report Manager. The replication mechanism is provided by Cronus, which is responsible for maintaining consistency and resolving conflicts between replicated object databases. Each Track Report Manager data object is a record of track information (track number, latitude, longitude, etc.) sent by the Parser Manager. The track information received by the Track Report Manager is either new or an update to an existing track object. For each new track, a new object is created and added to the database. For each track update, the Track Report Manager searches through its object database using the track number as the key and overwrites the already existing information stored in the track object.

```
24
   1 2680.00 9249.00  241.36 -99999.00 2700.00 9300.00 S F   0 0 N E  320    20.59 41.32  x
   2 2691.25 9221.75  261.49 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 56.09  x
   3 2700.25 9270.50  270.86 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 20.86  x
   4 2708.25 9241.00  281.55 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 42.48  x
   5 2678.25 9259.00  233.45 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 36.31  x
   6 2698.25 9252.25  267.31 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 33.82  x
   7 2680.00 9229.25  248.68 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 54.02  x
  10 2685.00 9242.50  229.76 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 53.81  x
  11 2727.50 9262.25  316.19 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 38.24  x
  12 2714.50 9213.50  283.87 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 62.73  x
  15 2666.00 9257.25  222.03 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 45.60  x
  14 2684.00 9253.25  222.98 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93 49.00  x
4232 2768.00 9198.00  314.20      0.00 2700.00 9300.00 A F   0 5 N E  159   240.30 98.60  x
4233 2679.25 9081.75  263.65      0.00 2700.00 9300.00 A F   0 5 N E   94   368.71 156.16  x
4234 2761.00 9107.25  295.44      0.00 2700.00 9300.00 A F   0 5 N E  126   382.64 148.20  x
4235 2811.75 9172.75  322.38      0.00 2700.00 9300.00 A F   0 5 N E  155   402.69 142.54  x
4236 2828.75 9249.00  344.94      0.00 2700.00 9300.00 A F   0 5 N E    0   393.75 133.52  x
4237 2801.50 9319.75    7.60      0.00 2700.00 9300.00 A F   0 5 N E  330   390.72 102.43  x
4240 2783.25 9144.50  308.39      0.00 2700.00 9300.00 A F   0 5 N E  141   399.73 136.84  x
4241 2800.50 9179.25  320.79      0.00 2700.00 9300.00 A F   0 5 N E  155   402.69 131.05  x
4242 2812.75 9214.00  332.57      0.00 2700.00 9300.00 A F   0 5 N E  172   397.75 127.62  x
4243 2785.00 9199.75  313.39      0.00 2700.00 9300.00 A F   0 5 N E  158   302.92  95.68  x
4244 2785.25 9244.25  335.79      0.00 2700.00 9300.00 A F   0 5 N E    0   337.50  93.72  x
4245 2675.25 9139.25  258.70      0.00 2700.00 9300.00 A F   0 5 N E  270   309.38 116.72  x
24
   1 2680.00 9249.00  241.36 -99999.00 2700.00 9300.00 S F   0 0 N E  320    20.59  41.32  x
   2 2692.25 9221.75  262.50 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  55.94  x
   3 2700.25 9270.50  270.86 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  20.86  x
   4 2708.25 9241.75  281.68 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  41.96  x
   5 2678.25 9259.00  233.45 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  36.31  x
   6 2699.25 9253.25  268.98 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  33.06  x
   7 2681.00 9229.25  249.67 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  53.64  x
  10 2666.00 9243.50  230.07 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  52.62  x
  11 2728.50 9262.25  317.22 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  38.97  x
  12 2715.50 9213.50  284.76 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  62.96  x
  15 2667.00 9258.25  222.20 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  44.38  x
  14 2685.00 9253.25  223.78 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  48.26  x
4232 2778.25 9194.00  316.44      0.00 2700.00 9300.00 A F   0 5 N E  159   240.30 106.33  x
4233 2679.25 9068.50  264.16      0.00 2700.00 9300.00 A F   0 5 N E   94   368.71 165.48  x
4234 2768.00 9095.75  296.64      0.00 2700.00 9300.00 A F   0 5 N E  126   382.64 158.32  x
4235 2823.75 9166.25  323.91      0.00 2700.00 9300.00 A F   0 5 N E  155   402.69 154.70  x
4236 2842.00 9249.75  346.53      0.00 2700.00 9300.00 A F   0 5 N E    0   393.75 146.20  x
4237 2813.75 9326.25    8.96      0.00 2700.00 9300.00 A F   0 5 N E  330   390.72 115.21  x
4240 2786.25 9142.00  308.96      0.00 2700.00 9300.00 A F   0 5 N E    0    28.13 140.01  x
4241 2810.75 9173.50  322.28      0.00 2700.00 9300.00 A F   0 5 N E    0    28.13 141.44  x
4242 2825.75 9210.75  334.31      0.00 2700.00 9300.00 A F   0 5 N E    0    28.13 140.17  x
4243 2778.25 9193.25  316.99      0.00 2700.00 9300.00 A F   0 5 N E  158   302.92 108.12  x
4244 2802.50 9244.25  339.58      0.00 2700.00 9300.00 A F   0 5 N E    0   337.50 109.61  x
4245 2675.25 9122.50  259.93      0.00 2700.00 9300.00 A F   0 5 N E  270   309.38 128.36  x
34
   1 2680.00 9250.00  240.87 -99999.00 2700.00 9300.00 S F   0 0 N E  320    20.59  40.70  x
   2 2692.25 9222.75  262.39 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  55.23  x
   3 2701.25 9270.75  273.63 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  20.73  x
   4 2708.25 9242.00  281.73 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  41.78  x
   5 2679.25 9260.00  234.05 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  35.15  x
   6 2699.25 9253.25  268.98 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  33.06  x
   7 2681.00 9229.50  249.60 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  53.48  x
  10 2666.00 9243.50  230.07 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  52.62  x
  11 2728.50 9262.50  317.41 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  38.84  x
  12 2715.50 9213.75  284.80 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  62.80  x
  15 2667.00 9258.25  222.20 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  44.38  x
  14 2685.00 9253.25  223.78 -99999.00 2700.00 9300.00 S F   0 6 N E  319    19.93  48.26  x
4232 2779.25 9192.50  317.17      0.00 2700.00 9300.00 A F   0 5 N E  159   240.30 109.20  x
```

Figure 2. Sample scenario file.

The Track Report Manager performs the seven operations that are described below:

InitializeDB The Track Report Manager manages two object types, TrackReportData and TrackTableIndex. TrackReportData is the type definition for the track objects that the Track Report Manager manages. TrackTableIndex is the type definition for the TrackTable, which is a list of track entries. Each track entry consists of a unique track number, an object UID that identifies the track object associated with the track number, and a version number that records the last time the track object was updated. The TrackTable is used as a reference to the track objects maintained by the Track Report Manager's object database. The InitializeDB operation removes the data in the object database of type TrackReportData to ensure a clean database when first running the Track Report

Manager. This operation also creates a new object for the type TrackTableIndex that stores the TrackTable. The UID of the TrackTable object is stored in Track Report Manager's generic as a reference to identify which object maintains the TrackTable. The InitializeDB operation must be invoked, via tropic, before any other operations are invoked on the Track Report Manager.

SetMonitorFlag For demonstration purposes, the Track Report Manager has a graphics capability that displays the flow of operation invocation from the Parser Manager and Graphics Map Client (figure 3). The SetMonitorFlag operation allows the user to activate the Track Report Manager's graphics by setting the flag to on via tropic. The flag is set to off by default.

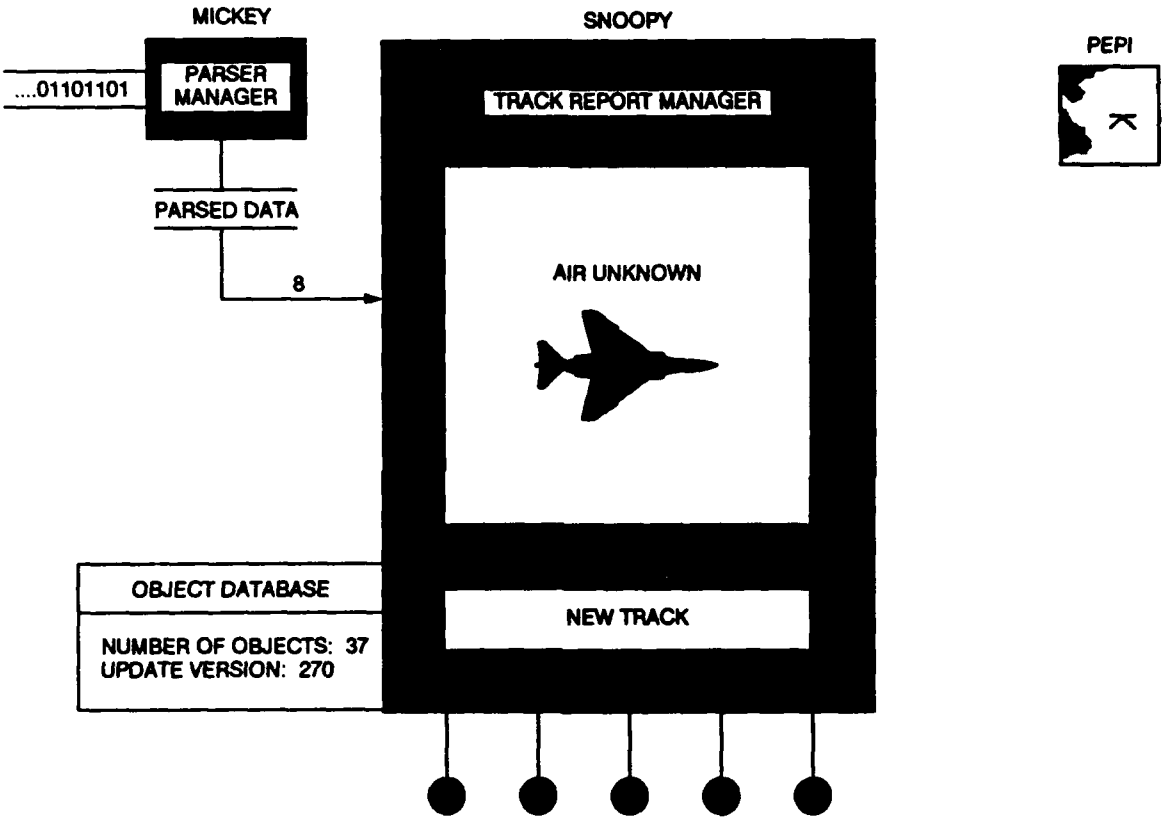

Figure 3. Track Report Manager graphical display.

TrackUpdate The Parser Manager passes an array of track information to the Track Report Manager by invoking the TrackUpdate operation. Each track is searched in the TrackTable (created by the InitializeDB operation) using the track number as the search key. If the track number is not found in the TrackTable, this operation will create a new object to store the new track information. The track number and the UID of the newly created object are entered in the TrackTable for future reference. If the track number is found in the TrackTable, the object UID associated with the track number is used to locate the track object in the Track Report Manager's database in which the new track information overwrites the old information. The version number in the TrackTable associated with this track number is incremented to reflect the update of the track object.

GetCurrentTracks The Graphics Map Client provides a graphics view of the track objects managed by the Track Report Manager. Track information is obtained by invoking the GetCurrentTracks

operation. The client maintains an epoch number that records the latest version of track information that the client has received. This epoch number is the parameter to the GetCurrentTracks operation that is used for comparison with the specific version number of each track entry in the TrackTable. An array of track information with version numbers greater than the epoch parameter is returned to the client. The client updates the epoch number for the next invocation. If the parameter to the operation is zero, all the track information in the Track Report Manager's database is returned. For debugging purposes, the operation can be invoked via tropic, which will textually display the contents of each track object. The information returned by the GetCurrentTracks operation consists of a track number, latitude and longitude of the track, bearing, latitude and longitude of the DLRP, altitude/depth, platform_type (e.g., aircraft, surface ship, subsurface ship), and category (friendly, hostile, unknown).

GetTrack The Graphics Map Client requires only the track number, latitude and longitude position of the track, altitude/depth, platform_type, and category information of each track to draw the proper icon on the map. For full information about a specific track, the GetTrack operation returns the information provided by the GetCurrentTracks operation, as well as additional information on track time, track quality, latitude and longitude direction, course, speed, range from DLRP, and nuclear identification. The Graphics Map Client invokes the GetTrack operation with the object UID of the track requested. The object UID is used to search the Track Report Manager's database for the information of the requested track. Similar to the GetCurrentTracks operation, this operation can be invoked via tropic for debugging purpose.

GetObjectDBStatus To monitor the status of replicated Track Report Managers, the Database Monitor Client invokes the GetObjectDBStatus operation, which returns the number of objects managed by each Track Report Manager and the epoch number of the last update of the objects in the database. The operation can also be invoked via tropic for debugging purpose.

GetTInfo For debugging purpose, the GetTInfo operation returns the information about the tasks currently running. The operation is invoked via tropic.

2.3 GRAPHICS MAP CLIENT

The Graphics Map Client is a worldwide dynamically scaled map package with the ability to display navy track symbols and other information. The map software is a modification of a map generator designed at the Information Sciences Institute, University of Southern California, that uses the Graphics Language device independent graphics package. This software was modified to use *SunCore*, a graphics package for Sun workstations, to construct the map from a digital map database. The software was originally designed as a situation assessment display for the Combat System Experiments project at NOSC. The model of the display was modified to run as a Cronus client with the capability to interface with a Cronus manager instead of interfacing with a centralized database.

The Graphics Map Client provides a graphical view of the object database maintained by the Track Report Manager (figure 4). Navy tracks stored in the object database are displayed in unclassified NTDS-like symbology. The scenario used to test and demonstrate the DOSE application is an outer air-battle scenario generated by the Interim Battle Group Tactical Trainer (IBGTT), which was recently renamed Research, Evaluation, and Systems Analysis (RESA). This scenario takes place in the Pacific Ocean with the DLRP at approximately 40 degrees latitude and 180 degrees longitude. The 74 tracks are a mixture of friendly, hostile, and unknown ships and aircrafts. Movement and modifications of the tracks in the Track Report Manager object database are reflected on the map. For example, if an unknown ship is later identified as a hostile ship, the icon will change to reflect the new identity.

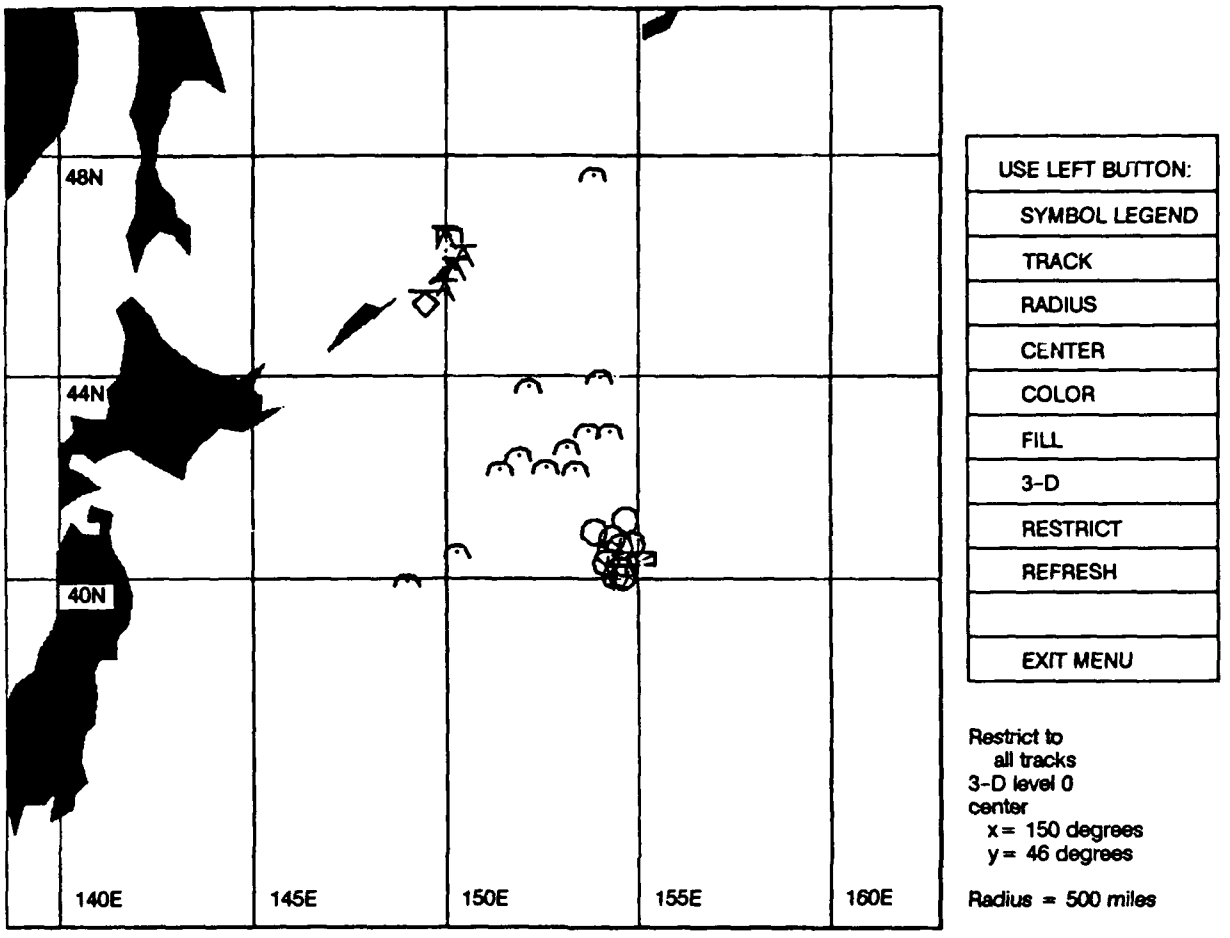

Figure 4. Graphics Map Client display.

The user can interactively manipulate the display via menus that are controlled by the mouse device. The following ten selections are available on the top-level menu:

1. Symbol legend - displays a symbol legend.
2. Track - allows the user to pick a track on the map (using the mouse) to display detailed information about the particular track. This selection invokes the GetTrack operation.
3. Radius - allows the user to change the radius of the map (in miles). This provides zooming and panning effects.
4. Center - allows the user to change the center (latitude, longitude) of the map (in degrees).
5. Color - allows the user to change the color (for color monitors) or pattern (for monochrome monitors) of the land on the map.
6. Fill - allows the user to select filled or outlined land forms on the map.
7. 3-D - allows the user to select the level of tilting of the map to provide a 3-dimensional projection.
8. Restrict - brings up a submenu for the user to select the type of tracks to display (for example, select only air friend tracks to be displayed on the map).

9. Refresh - redraws the screen display.

10. Exit - exits from the menu.

The Graphics Map Client automatically invokes the GetCurrentTracks operation every three seconds for the most recent track information.

2.4 DATABASE MONITOR CLIENT

The Database Monitor Client is an aid to Cronus' replication capability of the Track Report Manager. The client invokes Cronus' UpdateDaemon, which updates the replicated objects in the object database of all the Track Report Managers. This invocation is performed every 15 seconds by default, that is, the client sleeps for 15 seconds after each invocation. The user has the option to set the sleep period when the Database Monitor Client is activated. If the sleep period is set to a very low value (e.g., less than a second), the continuing invocation of the UpdateDaemon will take away processing time from other processes. If the sleep period is set to a large value (e.g., more than 10 minutes), it defeats the purpose of using the UpdateDaemon to ensure consistency of the object databases especially since the DOSE demonstration is only 20 minutes long. The invocation to the UpdateDaemon is necessary because the replicated Track Report Managers were not able to keep in sync with each other. This is due to the massive number of track updates the Parser Manager sends to the Track Report Manager. The Parser Manager sends an array of eight tracks to the Track Report Manager per invocation (*TrackUpdate*). After the Track Report Manager completes the operation, it propagates the eight changes to the other replicated Track Report Managers. This propagation may not complete because the Parser Manager will again invoke the Track Report Manager with new track updates.

The Database Monitor Client also acts as a debug aid. It broadcasts to all the replicated Track Report Managers and invokes the operation GetObjectDBStatus for the number of objects being managed by each manager. If the Track Report Manager graphics display is on, the number of objects will be displayed. Also displayed will be the updated version number that represents the number of updates performed by the Track Report Manager. These numbers provide a quick visual aid to show that the Track Report Managers are in sync with each other.

3.0 DOSE DEMONSTRATION

3.1 HARDWARE AND SOFTWARE CONFIGURATION

In order to demonstrate the DOSE application efficiently, the following display equipment and software are needed:

1. A minimum of two Sun 2 or Sun 3 workstations with a monitor (color or monochrome), that will be used for various graphics displays. Sun 386i workstations can be used to run the DOSE application without graphics displays. (The SunCore graphics package is not supported on the Sun 386i workstations.)

2. A microterminal (or equivalent) connected to each Sun workstation that will be used to activate graphics displays on the Sun monitor.

3. The **suntools** software package that will be used to provide multiple windows if hardware (Sun workstation and microterminal) availability is limited.

4. The **SunCore** graphics package that will be used by the DOSE managers and clients to provide graphics displays.

3.2 DEMONSTRATION STARTUP

The configuration layout of the DOSE application consists of four directories:

DBclient graphics parser trackrpt

The *DBclient* directory contains the **Database Monitor Client**. The *graphics* directory contains the **Graphics Map Client**. The *parser* directory contains the **Parser Manager** and the *trackrpt* directory contains the **Track Report Manager**.

The Parser Manager, Graphics Map Client, and Database Monitor Client all invoke the Track Report Manager. Thus, the Track Report Manager must be started first. The following describes how to start each manager and client to demonstrate the DOSE application.

3.3 TRACK REPORT MANAGER STARTUP

The Track Report Manager manages replicated data objects in its database. To demonstrate this feature, multiple copies of the Track Report Manager should be running on multiple host machines.

To bring up the first Track Report Manager, login as a Cronus user and go to the directory that contains the executable **trm** (*trackrpt* directory).

% cronus login cronus_user
% cd trackrpt

The Track Report Manager is a Cronus manager. Thus, the object database for its object type needs to be created. The syntax is */dbcreate=num* where *num* is the type number of the manager. The Track Report Manager manages two databases (types 321 and 322). By specifying *num* as 0, all appropriate databases are created for the manager.

% trm /dbcreate=0

This execution creates two files in the current directory: *objectdb.321* and *objectdb.322*.

To activate the manager, type the following command:

% trm /interactive

This executes the Track Report Manager in the foreground. Any messages from the manager will be printed onto the screen. The manager can also be activated in the background by replacing */interactive* with & to the above command line. Messages from the manager will be stored in the log.txt file.

The procedure for activating other Track Report Managers on other host machines (Cronus will not allow multiple copies of a Cronus manager to run on the same host machine), is similar to activating the first Track Report Manager, except, that */dbreplicate* is used instead of */dbcreate*.

% cronus login cronus_user
% cd trackrpt
% trm /dbreplicate=0
% trm /interactive

After the Track Report Manager(s) are up and running, use **tropic** to initialize one of the Track Report Manager's object database:

% cronus tropic
command: on {321}@**host** InitializeDB

@**host** is the hostname of the machine on which the Track Report Manager was first activated.

The Track Report Manager has graphics capability. To activate the graphics display, invoke the operation *SetMonitorFlag* and set the flag option through tropic:

% cronus tropic
command: on {321}@**host** SetMonitorFlag /flag

@**host** is the hostname of the machine on which the Track Report Manager is running. If there are multiple Track Report Managers running and the user wishes to activate graphics for all copies, **broadcast** can be used as the hostname.

To deactivate the graphics display, use the */noflag* option. By default, the option is set to */noflag*.

There are debug statements in the Track Report Manager that will output to the screen (where the manager was activated) when the manager is running. If the Sun Monitor is being used for the graphics display of the manager, it is best to activate the manager on a microterminal (or equivalent) connected to the Sun so that the debug statements will be printed on the microterminal. Otherwise, the debug statements will write on top of the graphics display.

3.4 PARSER MANAGER STARTUP

The Parser Manager manages one type of object. Unlike the Track Report Manager, it does not store information in its object database. The Parser Manager is a Cronus manager that functions as a client to the Track Report Manager. Duplicate copies of the Parser Manager (the Parser Manager is not a replicated manager like the Track Report Manager) can be running on multiple host machines.

However, for a demonstration purpose, usually only one Parser Manager is actively running and supplying track information to the Track Report Manager(s).

To activate the Parser Manager, login as a Cronus user and go to the directory that contains the executable **parser** (*parser* directory):

% cronus login cronus_user
% cd parser

The Parser Manager is a Cronus manager, thus, the object database for its type (this manager manages only one type, 320) needs to be created.

% parser /dbcreate=320

This execution creates 1 file in the current directory: *objectdb.320*.

To activate the manager, type the following command:

parser /interactive

The Parser Manager also has graphics capability. To activate the graphics display, invoke the operation *SetGraphicsFlag* and set the option flag through tropic:

% cronus tropic
command: on {320}@host SetGraphicsFlag /flag

To deactivate the graphics display, use the */noflag* option. By default, the option is set to */noflag*.

The unclassified version of the Parser Manager reads already parsed data stored in trackfile, an ASCII file. The operation ReadTracks was created for this purpose:

% cronus tropic
command: on {320}@host ReadTracks trackfile

3.5 GRAPHICS MAP CLIENT STARTUP

The Graphics Map Client is a Cronus client that invokes the Track Report Manager for track information. Unlike the Track Report Manager and the Parser Manager, the Graphics Map Client does not have the option of turning on or off its graphics display since its purpose is to graphically display the track information. Thus, the microterminal must be connected to the Sun workstation to activate the graphics display on the Sun Monitor (to take advantage of full screen display). If a microterminal is not available, the Graphics Map Client can be run in *GraphicsTool* of suntool window.

To activate the Graphics Map Client, go to the directory that contains the executable **graphics** (*graphics* directory) and startup the executable.

% cd graphics
% graphics

Successful activation of the Graphics Map Client will display a map on the monitor. A request to push the right button on the mouse flashes on the right-hand side of the screen. This action brings up a menu that allows the user to manipulate the map and view the tracks in different perspectives as described in section 2.3.

3.6 DATABASE MONITOR CLIENT STARTUP

The Database Monitor Client is a Cronus client that invokes the Track Report Manager for the number of objects in its object database. This client has no graphics display capability. However, the information the Database Monitor Client receives is displayed on the Track Report Manager graphics display.

To activate the Database Monitor Client, go to the directory that contains the executable **DBmonitor** (*DBclient* directory) and start-up the executable.

% cd DBclient
% DBmonitor

The Database Monitor Client invokes the Track Report Manager(s) in a continuous loop with a default sleep of 15 seconds between each invocation. The user can specify the sleep period by providing a numerical parameter to the *DBmonitor* executable. For example, if the user wants to set the sleep period to 30 seconds,

% DBmonitor 30

3.7 DEMONSTRATION SCRIPT

The following is a condensed demonstration script of the DOSE application as described earlier.

To start up the first Track Report Manager:

% cronus login
% cd trackrpt
% trm /dbcreate=0
% trm /interactive

To start up other Track Report Managers on other host machines:

% cronus login
% cd trackrpt
% trm /dbreplicate=0
% trm /interactive

After all the Track Report Managers are running, initialize the Track Report Manager on one of the host machines and activate the graphics display:

% cronus tropic
command: on {321}@host InitializeDB
command: on {321}@broadcast SetMonitorFlag /f
command: quit
%

Note: *@broadcast* will execute the *SetMonitorFlag* operation for all the Track Report Managers.

To start up the Parser Manager:

```
% cronus login
% cd parser
% parser /dbcreate=320
% parser /interactive
```

After the Parser Manager is running, start up the graphics display and read already parsed data in trackfile:

```
% cronus tropic
command: on {320}@host SetGraphicsFlag /f
command: on {320}@host ReadTracks trackfile
command: quit
%
```

To start-up the Graphics Map Client:

```
% cd /graphics
% graphics
```

To start-up the Database Monitor Client:

```
% cd /DBclient
% DBmonitor
```

3.8 DEMONSTRATION CLEANUP

The following steps will stop the services of the Track Report Manager and Parser Manager and perform the necessary cleanup. If the graphics displays of the Track Report Managers and Parser Manager are on, deactivate the graphics displays by using the option */n* in tropic for each manager. Use the Cronus command *stopservice* to stop the services of the managers. Enter each of the manager directories and remove the old object databases to ensure a clean start for the next demonstration.

```
% cronus tropic
command: on {321}@broadcast SetMonitorFlag /n
command: on {320}@host SetGraphicsFlag /n
command: quit
% cronus stopservice trm @host                    (on each of the machines)
% cronus stopservice parser @host
% cd trackrpt
% rm objectdb.321
% rm objectdb.322
% cd parser
% rm objectdb.320
```

The Graphics Map Client and Database Monitor Client are not Cronus managers. Thus, they are stopped by killing their process identifications (ids).

4.0 DEMONSTRATION TESTBED LAYOUT

There are many ways in which the DOSE application can be demonstrated. The method chosen to demonstrate DOSE will depend on hardware availability and the emphasis of the demonstration. The following describes some possible methods of demonstrating DOSE.

4.1 MINIMUM DEMONSTRATION LAYOUT

The minimum requirement needed to demonstrate the DOSE application is two Sun workstations. A microterminal connected to each of the Sun workstations would be advantageous. This would provide two extra monitors for the demonstration. A **suntools** package will provide multiple windows on the Sun Monitor and eliminate the use of microterminals. Within the suntools package is a **GraphicsTool** that can be used to activate graphics display of the managers and clients. The following is a possible layout to demonstrate DOSE:

Sun #1

1. Invoke suntools package.

2. Open two GraphicsTool windows. Use one window for the Track Report Manager and the second window for the Parser Manager.

3. Use the microterminal connected to the Sun for tropic use or open another shell window in suntools for the same purpose.

Sun #2

1. Invoke suntools package.

2. Open two GraphicsTool windows. Use one window for another Track Report Manager (to demonstrate replication) and the second window for the Graphics Map Client.

3. Use the microterminal connected to the Sun for activating the Database Monitor Client or open another shell window in suntools for the same purpose.

4.2 RECOMMENDED DEMONSTRATION LAYOUT

The ideal recommended layout to demonstrate the DOSE application would be six Sun workstations with a microterminal connected to each workstation. This would provide a full graphical view of the Track Report Manager, Parser Manager, and Graphics Map Client. The microterminals can be used as debug tools for displaying messages from the managers and client. Three Sun workstations would provide the opportunity to fully demonstrate replication with three Track Report Managers running concurrently. Load sharing could be demonstrated with two Graphics Map Clients invoking any one of the three Track Report Managers. Thus, the ideal layout would be the following:

three Track Report Managers on three different host machines,

one Parser Manager on one host machine, and

two Graphics Map Client on two different host machines.

Because the Database Monitor Client has no graphics display of its own, the display can be activated in the background on any one of the six host machines.

4.3 OTHER LAYOUTS

Many other possible layouts can be used to demonstrate the DOSE application. However, the minimum requirement is at least one Track Report Manager and one Parser Manager. The Graphics Map Client and Database Monitor Client are recommended but not required for the DOSE demonstration. The Track Report Manager's operations can be invoked through tropic rather than through the clients.

The current public unclassified distribution of the DOSE application is for Sun workstations only. The DOSE application is not limited to only Sun workstations. NOSC has demonstrated the DOSE application using Sun 2, Sun 3, Sun 386i, MicroVAX II, and IBM PC-AT.

5.0 REFERENCES

Berets, J. C., and R. M. Sands. January 1989. "Introduction to Cronus," Report No. 6986, BBN Systems and Technologies Corporation.

Gadbois, M. J. February 1990. "Distributed Operating System Experiment (DOSE) Application Installation Manual," NOSC Technical Document 1800.

Kwong (Gadbois), M. J., G. M. Sullivan, and L. C. Anderson. February 1989. "Distributed Operating System Experiment 1988 Source Code," NOSC Technical Note 1568.*

Sullivan, G. M., and L. Anderson. 31 August 1988. "The Distributed Operating System Experiment for FY87 (DOSE '87)," NOSC white paper.

*NOSC Technical Notes (TNs) are working documents and do not represent an official policy statement of the Naval Ocean Systems Center. For further information, contact the author.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1990	Final

4. TITLE AND SUBTITLE

DISTRIBUTED OPERATING SYSTEM EXPERIMENT (DOSE) APPLICATION USER'S MANUAL

5. FUNDING NUMBERS

PE: 0602232N
WU: DN888630

6. AUTHOR(S)

Manchi J. Gadbois

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Ocean Systems Center
San Diego, CA 92152-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NOSC TD 1801

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Technology Block Funding

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This document describes the functionalities and capabilities of the Distributed Operating System Experiment (DOSE) Version 4.0 application. Also described is how to run the application in various layouts according to the demonstration purpose. The companion to this document is "Distributed Operating System Experiment (DOSE) Application Installation Manual," NOSC Technical Document 1800.

14. SUBJECT TERMS

Cronus, Database Monitor Client, Track Report Manager, UNIX, Graphics Map Client, Parser Manager

15. NUMBER OF PAGES

22

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT