DTIC FILE COPY

(2)

Naval Ocean Research and Development Activity

August 1989 Report 214

Concept Definition: AN/SQQ-89 On-Board Trainer Scenario-Generation Software

AD-A222 460

DTIC
ELECTE
JUN 08 1990
S D
D

George J. Moss, Jr.
Ocean Technology Division
Ocean Acoustics and Technology Directorate

Approved for public release; distribution is unlimited. Naval Ocean Research and Development Activity, Stennis Space Center, Mississippi 39529-5004.

90 06 08 033

Foreword

This report describes a variety of software concepts developed by the Naval Ocean Research and Development Activity*. These computer-aided concepts should alleviate the labor-intensive operation when training scenarios for the AN/SQQ-89(V)-T() On-Board Trainer are generated and modified.

W B Moseley

J B Tupaz

W. B. Moseley
Technical Director

J. B. Tupaz, Captain, USN
Commanding Officer

* Now the Naval Oceanographic and Atmospheric Research Laboratory (NOARL).

Executive Summary

The AN/SQQ-89(V)-T() On-Board Trainer provides an at-sea training environment for sensor operators, ship teams, and ship/air teams by stimulating various sensors on surface ASW ships. Training sessions are controlled by scenario files, which specify the motions and the acoustic/electromagnetic emission parameters of simulated ships, submarines, and aircraft, as well as related medium propagation characteristics. Creating and modifying these scenario files is a labor-intensive operation. A variety of software concepts for computer-aided generation of training scenarios have been defined. A geometric transformation program permits the geometry of a scenario to be modified to support battle group training, to accommodate changes in sonar conditions, to optimize ship time and fuel usage, or simply to produce variety. A macrocompiler permits rapid assembly of a scenario from archived segments. A more advanced automatic macrocompiler permits the simulated targets and ownship to interact with each other according to user-defined functions during scenario compilation time. A reactive target emulator causes the simulated targets to react to each other and ownship in real time, thereby providing an interactive training environment.

DTIC COPY INSPECTED 6

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

Acknowledgments

The author is indebted to James Lennox of the TASC and to Elizabeth Powers of Planning Systems, Inc., who designed and coded the Scenario Transform Program. Douglas Clark of Analysis and Technology, Inc., and Mark Raphanelli and Buzz Campo of Raytheon Submarine Signal Division provided essential assistance by answering a variety of detailed questions about the On-Board Trainer scenario syntax. The effort was supported by the Naval Sea Systems Command, Surface Ship ASW Combat Systems Program Office (PMS411), Program Element 25620N. The author is especially indebted to J. H. Hammond, the NAVSEA Project Engineer, and to David Pope, recently retired from NAVSEA, for their guidance, suggestions, and encouragement during the investigation.

Contents

Concept Definition: AN/SQQ-89 On-Board Trainer Scenario-Generation Software

1.0 Introduction

This document describes a concept for software supporting off-line generation of scenarios for the AN/SQQ- 89(V)-T() On-Board Trainer (OBT).

The OBT and its scenario editing function are briefly described in Section 2. Section 3 outlines software concepts for simplifying the scenario generation process and making it less labor intensive. The functional capabilities of the software are described in Section 3.1, and the software's utilization and application are discussed in Section 3.2. Section 4 provides additional detail on the coordinate transformation and scenario merging software products. A brief summary is presented in Section 5.

2.0 Background

2.1 Description of the OBT

The AN/SQQ-89(V)-T() OBT, which is currently in production at Raytheon Submarine Signal Division, provides an on-board training environment for antisubmarine warfare (ASW) sonar operators, electronic support measures (ESM) operators, ship combat teams, and ship/air combat teams. It provides this training environment by stimulating the combat equipment, causing it to respond to simulated targets, the motions and characteristics of which are specified in a scenario file. For ASW training, inverse beamformed acoustic signals are injected into each stave channel of the AN/SQS-53 hull-mounted sonar, and into each hydrophone channel of the AN/SQR-19 towed array sonar. Simulated sonobuoy signals are injected into the AN/SQQ-28 via a cable to the LAMPS helicopter when the helicopter is on deck, or via radio link to the sonobuoy receiver when the helicopter is in the air. As a result, every console and electronic cabinet in the aforementioned sonars and in the Mk 116 ASW Control System behave as though the simulated targets are actually in the water.

For ESM training, display-level representations of target electromagnetic radiation are injected into the shipboard AN/SLQ-32 and the helicopter AN/ALQ-142 ESM sensors. This gives the equipment operators and combat teams practice in detecting, classifying, and reacting to electromagnetic threats, such as the tracking radars on hostile aircraft and their missiles.

The training process is controlled by an instructor's console. The instructor, however, relies heavily on preprogrammed scenario files, which specify the ocean environment, the passive and active acoustic characteristics of up to 10 ASW targets, the characteristics of the electromagnetic emitters on up to 100 ESM targets, and the initial positions and maneuvers of all targets. The scenario file is created and edited on the instructor's console, and is stored offline on a computer disk.

2.2 OBT Scenario Editing Facility Limitations

Scenario editing on the OBT instructor's console is accomplished one subevent at a time. For example, each change of a target's speed, course, altitude, or depth is specified separately. Therefore, the manual generation and modification of scenarios is a tedious, labor-intensive process.

Frequently, desired modifications to scenarios can be defined in terms of high-level concepts, such as geometric transformations to target tracks, the addition of predefined maneuvers to the tactics of the targets, or predefined interactive battle tactics, such as formations. The purpose of the software described in this document is to translate high-level definitions of platform positions and motions into detailed scenario code. Specific applications are discussed in Section 3.2.

3.0 Planned Software Capabilities

This section describes, in general terms, the capabilities of the proposed scenario generation software. Functional capabilities are described in Section 3.1. The manner in which these functional capabilities can be utilized for effective and efficient development of scenarios is described in Section 3.2. Some comments on the commonality among the various functional capabilities, from a technology standpoint, are offered in Section 3.3.

3.1 Functional Capabilities

The inputs to the off-line scenario generator are high-level instructions obtained through menu-driven interaction with the user, and (optionally) a scenario that has been created on the OBT and stored on a disk by the Scenario Input Computer (Unit 12). The off-line scenario generator creates a new scenario, or modifies an existing scenario, by adding and/or modifying subevents related to the positions and motions of targets, fixed bearing noise sources, and ownship. The output is a scenario file that can be loaded into the OBT and utilized for training.

The specific functional capabilities of the off-line scenario generator are geometric transformations, macro-compilation of individual events or sequences of events, automatic merging of sequences of events with parameter modification, graphical editing, and reactive target emulation. These functional capabilities are described in Sections 3.1.1 through 3.1.5. Application of these functional capabilities to the efficient and effective development of training scenarios is described in Sections 3.2.1 through 3.2.5.

3.1.1 Geometric Transformations: The Scenario Transform Program performs geometrical transformations on existing scenarios. The input is a complete scenario, with platform motions defined. The Scenario Transform Program can rotate, translate, expand, or contract individual tracks or the entire scenario. The output is a modified scenario file that can be executed by the OBT. Graphical displays of track plots are provided.

Expansion or contraction can be accomplished by changing subevent times while holding platform speeds constant, or by changing platform speeds while holding subevent times constant. However, a constant speed expansion or contraction cannot be performed on individual tracks. Requested expansions and contractions are error-checked for violations of platform speed limits and for the ability of the platforms to complete course changes. The specific functions of the Scenario Transform Program are discussed in Section 4.1. Applications of the Scenario Transform Program are discussed in Section 3.2.1.

3.1.2 Scenario Macrocompiler: The Scenario Transform Program performs operations on a complete scenario or on individual tracks. The next logical extension of this capability is software to perform operations on individual events. These operations include changes of parameters, deletions, and insertions of events. The insertion function is powerful enough to insert sequences of events contained in a separate file. Therefore, the user is able to create libraries of scenario segments. Scenario segments stored in these libraries can then be assembled quickly and efficiently into complete scenarios. Detailed applications are described in Section 3.2.2. A detailed functional description is presented in Section 4.2.

3.1.3 Automatic Merging of Scenario Segments: Automatic merging of scenario segments is an extension of the scenario macrocompiler concept, wherein the decision to insert a sequence of subevents from a secondary file is based on specifiable functions of scenario platform positions, speeds, courses, and emissions. Also, the parameters of subevents in these secondary files are allowed to be variables that are functions of scenario platform positions, speeds, and courses.

The automatic merging function operates by stepping through a scenario, incrementing problem time. Whenever problem time is incremented, the positions, courses, and speeds of ownship and all targets, and the pinging status of ownship, are calculated. User-defined conditional functions of these calculated variables are evaluated. A sequence of subevents stored in a secondary file is inserted whenever one of these conditional functions is satisfied.

Once a sequence of subevents is inserted, user-defined functions specifying the subevent parameters contained therein are also evaluated whenever problem time is incremented.

The most critical task in the development of the automatic merging function is the design of a simple, straightforward, user-friendly technique for invoking it. This task is accomplished through the use of pseudoevents that are inserted into the scenario by the user. For example, assume that the user wishes a target submarine to perform passive target motion analysis on another target, followed by an attack. The operator uses editing functions to insert a non-OBT-readable pseudoevent into the scenario. The automatic merging function parses that pseudoevent, and loads the necessary control functions from a library. Execution of these functions causes events specifying the motions and activities of the attacker to be adjusted for the motions of its target, and to be inserted into the scenario.

Typical applications of automatic merging of scenario segments are described in Section 3.2.3.

3.1.4 Graphical Editor: The graphical editor is an enhancement to the scenario macrocompiler function. It is also compatible with the automatic merging of scenario segments function.

The graphical editor function allows the user to draw platform tracks on the computer screen with a mouse. The user uses the mouse to place a cursor on a track and to hook the track by depressing the left mouse button. When he moves the mouse, the point on the track which he hooked follows the cursor like a stretched rubber band. Depressing the right mouse button then causes the new temporary shape of the track to become permanent. The user is then asked whether he wishes to accept the default assumption that the recent platform speed has been constant. If he does not accept this assumption, then he is asked to resolve the ambiguity between the time of the new turn and the speed during the previous leg. A subevent for the newly defined turn is then inserted into the scenario, and any necessary modifications to previous and following subevents are made.

In the event the user hooks a turn, rather than a leg of the scenario, the position of the turn is moved, and a new subevent is not inserted.

Repeated invocation of this function permits complex tracks to be created in a simple and straightforward manner. Advantages are discussed in Section 3.2.4.

3.1.5 Reactive Targets: The automatic merging of scenario segments permits the definition of targets that can react to the position, course, and emissions of ownship and other targets during off-line scenario generation. The next logical extension of this capability is to provide targets with the ability to react to the trainees' actions in real time. This reaction can be accomplished by taking the following developmental steps: (a) create a data channel that inputs real-time data and makes it available to the automatic merging function; (b) drive the automatic merging function with real time rather than simulated platform parameters; and (c) provide a scenario parser and a data channel, which enables the scenario segments loaded and modified by the automatic merging function to control the OBT in real time. Once this is accomplished, the OBT is capable of real-time dynamic interaction with the trainees. Applications of this capability are discussed in Section 3.2.5.

Two major issues must be resolved before a reactive target capability can be implemented. The first issue is whether the reactive target controller must control only those targets which the operator designates as reactive, or whether it will be required to control the entire scenario. Present OBT protocols lock out the OBT's scenario parsing capability for the duration of the scenario, whenever an external controller takes control. The alternatives that should be considered are (a) modifying the protocols to allow coordinated shared control between the OBT and an external controller, and (b) duplicating a substantial part of the OBT's scenario parsing capabilities.

The second issue is the selection of the physical channel for communication between the reactive target controller and the OBT. The alternatives are to (a) provide the hardware and software for use of the OBT's external control port, and (b) modify the OBT software itself to channel the necessary messages through the channel between the OBT's Signal Generation and Processing unit (Unit 2) and its Scenario Input Computer (Unit 12).

3.2 Applications and Examples

The following paragraphs describe situations in which a high-level, off-line scenario modification and generation facility can significantly enhance the ability of the surface ASW ships to utilize the OBT.

3.2.1 Geometric Transformations: This section discusses applications of the Scenario Transform Program, which is discussed in Section 3.1.1.

When sonar operators are being trained to work with barely visible signals, it is often desirable to adjust the ranges to targets to accommodate a change in the sonar range of the day. A simple expansion or contraction in the scale of the scenario is more manpower efficient than reprogramming the entire scenario.

Time and fuel can be saved if ownship can make progress toward its next port, station, or patrol area while prosecuting simulated targets. Unfortunately, hostile targets in preprogrammed scenarios are not always positioned in that direction. The Scenario Transform Program provides a simple means to rotate the individual tracks or the complete scenario, thereby placing targets closer to the desired direction for ownship movement.

Multiple-ship team training can be accomplished if a single scenario is adjusted through translation of all tracks to compensate for the initial position of each ship. In this way, an entire battle group can practice prosecuting the same simulated targets.

Variety can be introduced to counteract the tendency of trainees to memorize exercises through selective translation, rotation, scale expansion, and scale contraction of individual tracks.

3.2.2 Scenario Macrocompiler: The principal use of the scenario macrocompiler is to reduce the amount of repetitive keypunching needed to create a scenario. It allows the user to create frequently used sequences of subevents, store them off-line, and merge them into scenarios when needed. Therefore, its contribution to scenario development is similar to the contribution of the macroassembler to software development. Like the macroassembler, the productivity enhancement of the macrocompiler will depend upon how well the user organizes his scenario development to take advantage of repetitive sequences of subevents.

Examples of subevent sequences, which can be created in advance and stored off-line in macros, include the following: initial setup of platforms, sensors, and emitters; passive target motion analysis maneuvers; baffle clearing operations in which a submarine periodically circles to check for threats behind him; and formations.

The scenario macrocompiler is limited because the merged subevent sequences are not automatically adjusted for the context of the scenario. Thus, post-merge editing is required to make inserted targets track, attack, or evade other targets.

3.2.3 Automatic Merging of Scenario Segments: The automatic merging function permits prior creation, off-line storage, and subsequent insertion into a scenario of subevent sequences that involve target reactions to position, speed, and emissions of another target or ownship. Thus, targets can be programmed to react automatically to other targets or ownship during scenario compilation time. Tactically realistic scenarios can then be programmed rapidly using techniques described in the following text. Use of the automatic merging function to create a limited training environment of fronts and eddies is also described.

A ***torpedo macro*** can be defined with the following specified behavior: (a) initially, maintain a position identical to that of a designated attack submarine and remain invisible to sonar, (b) become enabled upon closure to within a designated range of a specified victim, which may be another target or ownship, (c) proceed toward the victim and enter a search pattern, and (d) upon satisfaction of acquisition criteria, attack.

• The *attack submarine* in the previous paragraph can be programmed to (a) execute a specified search pattern, (b) initiate prescribed target motion analysis maneuvers upon satisfaction of acquisition criteria for the designated victim, and (c) steer an intercept course while performing any maneuvers required to maintain the target motion analysis with passive sonar.

• *Individual ships* or *aircraft* can be programmed to approach another platform and to fall into formation at a designated range and relative bearing, upon intercept. Thus, groups of ships or aircraft can be programmed to fall into formation and follow a leader.

• *Aircraft* can be loaded with missiles and steered as a unit. This maneuver is significant because the present OBT scenario programming language does not contain the concept of a loaded aircraft; the scenario developer is responsible for steering each missile independently, even if it is disabled and attached to the wing of an aircraft.

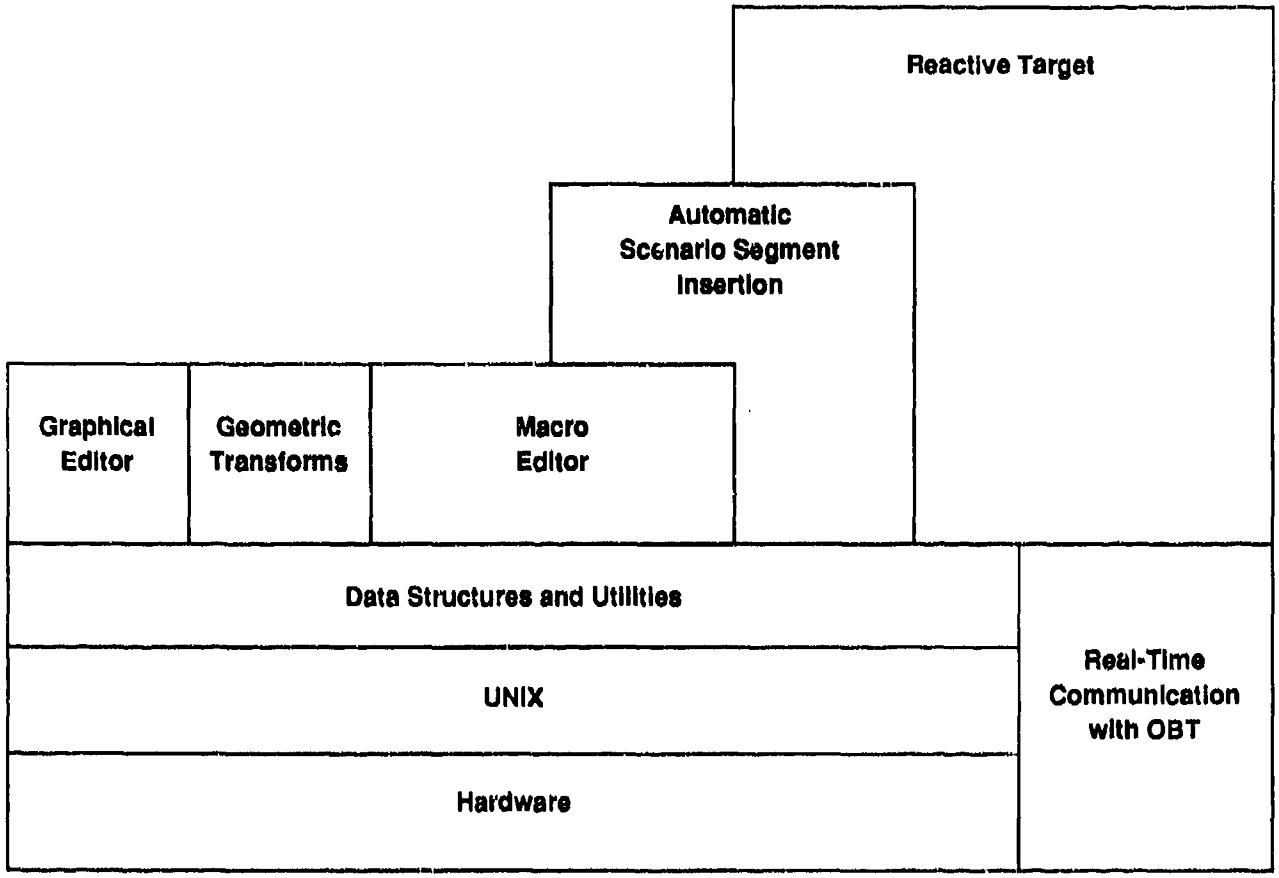
•Development of *complex ESM scenarios* can be significantly simplified through the use of macros that define multiple-target attack patterns for aircraft and their missiles. The ESM environment for a major air attack against a convoy or battle group can then be programmed by specifying initial positions of attacking aircraft and by assigning their targets.

A ***limited, nonhomogeneous ocean training environment*** for ownship sensors can be created by combining the automatic merging function with a specialized acoustic model. The requirements for the acoustic model are discussed, as well as how the automatic merging function could be used to inject the nonhomogeneous acoustic behavior of the ocean into the training environment.

• The acoustic model can be prepared by analyzing a particular oceanographic regime (such as a front or eddy in a specific ocean) for propagation losses. This analysis would involve the use of software that is not a part of the scenario generation software. Subtraction of loss computed by the OBT's propagation model (for a specified ocean) from the propagation loss computed by the nonhomogeneous ocean acoustic model would yield the necessary corrections to the OBT's propagation model. For rapid scenario compilation, the numerical acoustic analysis results would have to be precomputed and stored as a data base, in which propagation loss corrections could be obtained by interpolation between stored values. A typical data base might divide the exercise area into a 20 x 20 horizontal matrix with five possible depths. A data base of this resolution would require storage of 800,000 propagation loss corrections (based on 400 possible positions for a surface ship times 2000 possible positions for a submarine).

• During scenario compilation, the problem geometry of the scenario would be simulated. The specific positions of ownship and all acoustic targets would be used as the basis for acquisition of propagation loss corrections from the acoustic data base. Propagation loss corrections would be inserted into the scenario when they differ from previously inserted values by a prescribed threshold.

— Injection of propagation loss corrections into the scenario can be done by inserting subevents with the action tokens TTNPASUS (contact passive strength) and TTNACTUS (contact active strength). These corrections enable the program to provide a dynamic range of 40 dB for passive signals and 24 dB for active signals. It also consumes subevents, thereby reducing the tactical complexity of the scenario.



System structure.

— Sonobuoys could be accommodated if the sensor injection levels and the target active and passive strengths were carefully balanced to simulate the correct propagation loss for all target-to-sensor paths. However, since the AN/SQQ-28 simulation function has only one injection level control, all sonobuoys simulated at a particular time would have to be located in the same area.

3.2.4 Graphical Editor: The graphical editor allows the user to think directly in terms of track plots when laying out the tracks. This capability results in enhanced productivity and easier use of all applications.

3.2.5 Reactive Targets: A reactive target capability allows the applications discussed in Section 3.2.3 to run in real time, rather than only during scenario compilation. Team trainees benefit from the experience of dealing with targets that react to their maneuvers and actions.

In addition, instructors are able to inject real-time maneuvers of ESM targets, such as the deflection of an incoming missile, by chaff or jamming. The pace of ESM evolutions is simply too fast to permit effective manual reactive control in real time with existing techniques.

3.3 System Structure

The figure shows the overall structure of the software described in this report. The foundation for all software items described in Section 3.1 is a set of data structures and utilities for access to manipulate and display scenario data. Therefore, the development of each of the software items discussed will make a contribution to the software infrastructure, which will be exploited by the next phase of development. In addition, the automatic scenario segment insertion program (Sect. 3.1.3) will build on and expand the macroeditor (Sect. 3.1.2). The reactive target controller builds on the automatic scenario segment insertion program, but requires an additional capability for real-time communication with the OBT.

4.0 Detailed Functional Descriptions

4.1 Scenario Transform Program

The Scenario Transform Program has five specific functions.

• The scenario file is loaded and is restructured to permit easy access to individual tracks.

• If desired, the tracks can be displayed graphically.

• The graphical display can be zoomed to permit large-scale viewing of a portion of the scenario.

• Operator requests for translations, rotations, expansions, and contractions are error-checked, and the necessary operations are performed.

• A file name consistent with OBT conventions is established. The subevents are sorted into chronological order and are recorded into the new file in a format, which the OBT can read and execute.

4.2 Scenario Macrocompiler

The following software functions are provided by scenario macrocompiler:

4.2.1 Load Scenario: The load scenario function loads the scenario file. In addition it restructures the scenario to permit easy access to individual tracks and subevents. Individual subevents are accessible through a sequence of pointers that support efficient insertions and deletions.

4.2.2 Display and Zoom: The display and zoom function allows graphical display of the the track plots. The user can zoom the display for detailed viewing.

4.2.3 Script Display: The user must point to specific events to make a change or insertion. A script display function is provided for this purpose. The script display function examines the appropriate page of the scenario, parses the subevents, merges subevents into events, and displays the script in a display format similar to that used by the OBT.

Pointers to event numbers can be controlled through arrow keys or soft keys. The event numbers being pointed to are highlighted through inverse video, underlining, or display of a cursor. When either the up and down arrow keys or the soft keys are pressed, the pointer moves from event to event. Optional parallel control through a mouse is desirable.

Keyboard entry of an event number causes (a) positioning of the pointer on that event, (b) highlighting of that event, and (c) display of the appropriate page of script, if different from the page currently being displayed.

The following soft keys are displayed when the script display is active: page forward, page back, delete event, change event, and add event.

• The ***page forward*** soft key causes the next page of the script to be displayed.

• The ***page back*** soft key causes the previous page of the script to be displayed.

• The ***delete*** soft key causes the user to be prompted with an "Are you sure?" message. If, and only if, the user confirms his intent to delete, the event being pointed to and highlighted is deleted. Appropriate pointers in the data structure are modified as required to bypass the event being pointed to. The displayed script is rewritten without the event.

• The ***change event*** soft key prompts the operator for an event number. The event currently being pointed to on the script display page is the default, if the operator types a carriage return without first entering an event number.

The screen then displays selected status information plus a menu. The menu items are based on the menu selections available to an operator of the OBT in the scenario generation mode under comparable circumstances. However, reduced screen size relative to the OBT's trainer control console may necessitate splitting the menu into several pages.

Event time is displayed and is defaulted to the time of the selected event. All menu selections are processed in approximately the same sequence and under approximately the same conditions as they would be in the scenario generation mode of the OBT; corresponding changes occur in the scenario script. These changes include modification of all appropriate pointers in the data structure and renumbering of all events affected by any change in the event time.

The scenario script, updated to reflect the changes, is displayed. The changed event continues to be highlighted, and any page changes necessary to display this event are executed.

• The ***add event*** soft key causes a menu display to offer the choice of either a single event or a sequence of events, which are stored in a separate macrofile.

If the single event option is selected:

> An event is created in the next available memory space in the scenario data structure. The user is then prompted for the type or menu level of the event (such as ownship or contacts).
>
> The screen displays selected status information plus a menu. The menu occupies as many pages as are required to fit the information on the available display screen. The menu items are based on the menu selections available to an operator of the OBT in the scenario generation mode under comparable circumstances. For example, if the menu level of the selected event is "ownship," the available menu items include motion, ordered course, ordered speed, ordered rudder angle, AN/SQS-53, AN /SQR-19, AN/SQQ-28, and change event time.
>
> Event time is displayed and is defaulted to zero. Selecting "change event time" results in a prompt for the time in hours:minutes:seconds, with the default being the time of the event previously pointed to on the script page. All other menu items are processed in basically the same sequence and under basically the same conditions as they would be in the scenario generation mode of the OBT, and result in corresponding changes to the scenario script. These changes include modification of all appropriate pointers in the data structure, and renumbering of all events as required to make all events in the scenario occur in time sequence. The script display, appropriately updated, is rewritten. The added event is highlighted, and the page of the script on which this event appears is displayed.

If the option to insert a sequence of events stored in a macrofile is selected:

> The operator is prompted for an event time. The default is the time of the event currently being pointed to.
>
> A list of appropriate macrofiles is displayed. The operator is prompted to select a specific macrofile. The selected file, which contains subevents in the standard file server format, is loaded into memory and integrated into the scenario data structure. Pointers in the data structure are adjusted to position the sequence of events appropriately for the selected event time. The selected event time is then added to the event times in the new data; therefore, the event times in the inserted file are interpreted as incremental times relative to the selected event time. Other subevent parameters are either adjusted or passed without change based on the following criteria: (a) rules determined during design of the macrocompiler, (b) specifications stored in the header record of the macrofile, and (c) operator interaction where appropriate. The scenario is sorted, so all events appear in the proper time sequence. Events are numbered sequentially throughout the scenario. The script display, appropriately updated, is rewritten. The first event of the inserted sequence is highlighted, and the page of the script on which this event appears is displayed.

4.2.4 Geometrical Transformation Functions: Operator requests for translations, rotations, expansions, and contractions of the entire scenario or individual tracks, are accepted and processed as in the Scenario Transform Program.

4.2.5 Generate Scenario: A file name consistent with OBT conventions is established. The subevents are sorted into chronological order and are recorded into the new file in a format which the OBT can read and execute. Event numbers increase sequentially throughout the scenario. The maximum file size is 6000 subevents.

4.2.6 Generate Subevent Sequence File: The generate subevent sequence file function, through dialog with the operator, establishes a macrofile name consistent with OBT conventions. An initial character (to be determined) is used to distinguish this macrofile from scenarios and other files used by the OBT. The user is given the opportunity to specify whether selected classes of subevent parameters are to be treated as absolute or relative upon macro insertion; user decisions are logged in the header field of the macrofile. The subevents are then sorted into chronological order, and are recorded into the new file in a format which can be read to the "add event" function by the "sequence of events" option described in section 4.2.

5.0 Summary

The subevent-by-subevent syntax of the AN/SQQ-89 OBT's scenario editor makes the preparation of scenario scripts a labor-intensive operation. As a result, off-line scenario generation software is needed.

A variety of software concepts for off-line scenario generation has been presented. Functional capabilities have been described for software that supports geometric transformations, macrocompilation of individual subevents or sequences of subevents, automatic merging of sequences of subevents with parameter modification, graphical editing, and reactive target emulation. Examples of applications illustrate the software's capability for simplifying the scenario development process and enhancing OBT utilization.

Distribution List

Asst Secretary of the Navy
(Research, Engineering & Systems)
Navy Department
Washington DC 20350-1000

Chief of Naval Operations
Navy Department (OP-02)
Washington DC 20350-2000

Chief of Naval Operations
Navy Department (OP-71)
Washington DC 20350-2000

Director
National Ocean Data Center
WSC1 Room 103
6001 Executive Blvd.
Attn: G. W. Withee
Rockville MD 20852

Chief of Naval Operations
Navy Department (OP-987)
Washington DC 20350-2000

Oceanographer of the Navy
Chief of Naval Operations
Attn: OP-096
U.S. Naval Observatory
34th & Mass Ave., NW
Washington DC 20390-1800

Commander
Naval Air Development Center
Warminster PA 18974-5000

Commanding Officer
Naval Coastal Systems Center
Panama City FL 32407-5000

Commander
Space & Naval Warfare Sys Com
Washington DC 20363-5100

Commanding Officer
Naval Environmental Prediction
Research Facility
Monterey CA 93943-5006

Commander
Naval Facilities Eng Command
Naval Facilities Eng Command Headquarters
200 Stovall St.
Alexandria VA 22332-2300

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 100
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 125L (13)
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 125P
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 105
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 115
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 200
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 252
Stennis Space Center, MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 300
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Research Laboratory
Washington DC 20375

Commander
Naval Oceanography Command
Stennis Space Center MS 39529-5000

Commanding Officer
Fleet Numerical Oceanography Center
Monterey CA 93943-5005

Commanding Officer
Naval Oceanographic Office
Stennis Space Center MS 39522-5001

Commander
Naval Ocean Systems Center
San Diego CA 92152-5000

Commanding Officer
ONR Branch Office
Box 39
FPO New York NY 09510-0700

Commander
David W. Taylor Naval Research Center
Bethesda MD 20084-5000

Commander
Naval Surface Weapons Center
Dahlgren VA 22448-5000

Commanding Officer
Naval Underwater Systems Center
Newport RI 02841-5047

Superintendent
Naval Postgraduate School
Monterey CA 93943

Director of Navy Laboratories
Rm 1062, Crystal Plaza Bldg 5
Department of the Navy
Washington DC 20360

Officer in Charge
New London Laboratory
Naval Underwater Sys Cen Det
New London CT 06320

Director
Office of Naval Research
Attn: Code 10
800 N. Quincy St.
Arlington VA 22217-5000

Director
Woods Hole Oceanographic Inst
P.O. Box 32
Woods Hole MA 02543

University of California
Scripps Institute of Oceanography
P.O. Box 6049
San Diego CA 92106

Officer in Charge
Naval Surface Weapons Center Det
White Oak Laboratory
10901 New Hampshire Ave.
Attn: Library
Silver Spring MD 20903-5000

Commanding Officer
Fleet Anti-Sub Warfare Training Center, Atlantic
Naval Station
Norfolk VA 23511-6495

Brooke Farquhar
NORDA Liaison Office
Crystal Plaza #5, Room 802
2211 Jefferson Davis Hwy.
Arlington VA 22202-5000

Director
Defense Mapping Agency Sys Cen
Attn: SGWN
12100 Sunset Hill Rd. #200
Reston VA 22090-3207

NORDA
Code 125EX
Stennis Space Center MS 39529-5004
(Unlimited only)

Director
Office of Naval Technology
Attn: Dr. P. Selwyn, Code 20
800 N. Quincy St.
Arlington VA 22217-5000

Director
Office of Naval Technology
Attn: Dr. C. V. Votaw, Code 234
800 N. Quincy St.
Arlington VA 22217-5000

Director
Office of Naval Technology
Attn: Dr. M. Briscoe, Code 228
800 N. Quincy St.
Arlington VA 22217-5000

Director
Office of Naval Research
Attn: Dr. E. Hartwig, Code 112
800 N. Quincy St.
Arlington VA 22217-5000

Director
Office of Naval Research
Attn: Code 12
800 N. Quincy St.
Arlington VA 22217-5000

Director
Office of Naval Research
Attn: Dr. E. Silva, Code 10D/10P
800 N. Quincy St.
Arlington VA 22217-5000

Chief of Naval Operations
Navy Department (OP-0962X)
Attn: Mr. R. Feden
Washington DC 20350-2000

Commander
Naval Sea Systems Command
Naval Sea Systems Command Headquarters
Washington DC 20362-5101

Commanding Officer
Naval Civil Engineering Laboratory
Port Hueneme CA 93043

Commander
Naval Air Systems Command
Naval Air Systems Command Headquarters
Washington DC 20361-0001

Pennsylvania State University
Applied Research Laboratory
P.O. Box 30
State College PA 16801

University of Texas at Austin
Applied Research Laboratories
P.O. Box 8029
Austin TX 78713-8029

Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Rd.
Laurel MD 20707

University of Washington
Applied Physics Laboratory
1013 Northeast 40th St.
Seattle WA 98105

Commander
Naval Sea Systems Command
Attn: G. Moss (PMS411241)
Washington DC 20362-5101

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 250
Stennis Space Center MS 39529-5004

Commanding Officer
Naval Ocean R&D Activity
Attn: Code 252 (12)
Stennis Space Center MS 39529-5004

Tracor (OBT Library)
Crystal Park 3, Suite 600
2231 Crystal Dr.
Arlington VA 22202

Commander
Naval Underwater Systems Center
Attn: Code 3123 (C. Cable)
New London Laboratory
New London CT 06320-5594

Commander
Naval Underwater Systems Center
Attn: Code 411 (R. Bucettas)
New London Laboratory
New London CT 06320-5594

Commander
Naval Sea Systems Command
Attn: J. Hammond (PMS411312)
Washington DC 20362-5101

Commander
Naval Underwater Systems Center
Attn: Code 3123 (L. Mangiafico)
New London Laboratory
New London CT 06320-5594

Commander
Naval Underwater Systems Center
Attn: Code 3123 (J. Wachter)
New London Laboratory
New London CT 06320-5594

Analysis & Technology, Inc.
Attn: D. L. Clark
Two Crystal Drive, Eighth Floor
Arlington VA 22202

Analysis & Technology, Inc.
Attn: T. Abbott
Technology Park, Route 2
P.O. Box 220
North Stonington CT 06359

Analysis & Technology, Inc.
Attn: D. L. Holloway
Technology Park, Route 2
P.O. Box 220
North Stonington CT 06359

Planning Systems, Inc.
Attn: E. Powers
115 Christian Lane
Slidell, LA 70458

Commanding Officer
Naval Research Laboratory
Attn: Code 5510 (Dr. R. Schumacher)
Washington DC 20375

Commanding Officer
Naval Research Laboratory
Attn: Code 5555 (J. Molnar)
Washington DC 20375

Raytheon Company
Submarine Signal Division
Attn: Joseph Ricci (4)
P.O. Box 360
Portsmouth RI 02871

TASC
Attn: J. E. Lennox
55 Walkersbrook Dr.
Reading MA 01867

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

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

1. Agency Use Only *(Leave blank).*

2. Report Date. August 1989

3. Report Type and Dates Covered.

4. Title and Subtitle.
Concept Definition: AN/SQQ-89 On-Board Trainer Scenario-Generation Software

5. Funding Numbers.
Program Element No. 25620N
Project No. 00101
Task No. 101
Accession No. 92528A

6. Author(s).
George J. Moss, Jr.

7. Performing Organization Name(s) and Address(es).
Ocean Acoustics and Technology Directorate
Naval Ocean Research and Development Activity
Stennis Space Center, Mississippi 39529-5004

8. Performing Organization Report Number.
NORDA Report 214

9. Sponsoring/Monitoring Agency Name(s) and Address(es).

10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited. Naval Ocean Research and Development Activity, Stennis Space Center, Mississippi 39529-5004.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

The AN/SQQ-89(V)-T( ) On-Board Trainer provides an at-sea training environment for sensor operators, ship teams, and ship/air teams by stimulating various sensors on surface ASW ships. Training sessions are controlled by scenario files, which specify the motions and the acoustic/electromagnetic emission parameters of simulated ships, submarines, and aircraft, as well as related medium propagation characteristics. Creating and modifying these scenario files is a labor-intensive operation. A variety of software concepts for computer-aided generation of training scenarios have been defined. A geometric transformation program permits the geometry of a scenario to be modified to support battle group training, to accommodate changes in sonar conditions, to optimize ship time and fuel usage, or simply to produce variety. A macrocompiler permits rapid assembly of a scenario from archived segments. A more advanced automatic macrocompiler permits the simulated targets and ownship to interact with each other according to user-defined functions during scenario compilation time. A reactive target emulator causes the simulated targets to react to each other and ownship in real time, thereby providing an interactive training environment. A schedule for implementation of the software concepts is suggested.

14. Subject Terms.
expert systems, knowledge systems, artificial intelligence

15. Number of Pages. 10

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	None

NSN 7540-01-280-5500

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