Copy No: _____

User Interface Specification and Functional Notes for the Dynamic Air Threat Simulator Scenario Editor

The DATS Scenario Editor

Marc Grushcow
NTT SYSTEMS INC

DRDC – Toronto

Contractor's Report

[Enter document number here]

2003-12-22

Author

Grushcow, Marc

Approved by

[Enter name here]
[Enter title here]

Approved for release by

[Enter name here]
[Enter title here]

[Enter admin info here]

[Enter standard clause here]

Abstract

DATS, the Dynamic Air Threat Simulator, is a task designed to support investigations into how experts use cues in naval air defence environments. Much of its effectiveness depends on presenting subjects with realistic and challenging scenarios. This document presents a detailed description of the DATS scenario editor user interface. This allows authors to place multiple targets on a map underlay and control their behaviour in time and space.

Scenario authors will have detailed control over target properties and these can change as the scenario unfolds. The editor itself however, treats targets simply as moving objects that have properties, most of which it merely maintains on behalf of the author. This insensitivity to the semantics of most properties allows it to server a variety of environments. The current version supports DATS and treats terrain as flat and featureless; a simplification that fits well with naval air defence problems. A more sophisticated terrain model could accommodate land forces problems, and support for roads and determining visibility would make the editor suitable for urban situations.

The scenario editor is implemented as a standard Windows application using leading edge technologies. This makes it compatible with Microsoft's current and planned desktop systems. The interface is consistent with office applications and users should find it easy to learn and use.

Résumé

[Enter text: French]

This page intentionally left blank.

Table of contents

List of figures

This page intentionally left blank.

Introduction

This document is a detailed user interface specification for the DATS Scenario Editor. DATS, the Dynamic Air Threat Simulator, is a system designed to support investigations into how experts use cues in naval air defence environments. The functional specifications for the task, the scenario editor, and data analysis facilities are described in a separate report[1].

The scenario editor allows authors to place targets on a map and design flight plans that they will execute when the task is run. The targets and their behaviour are intended to create a realistic and challenging environment.

Although created for DATS, the scenario editor will be designed in a general way based on the broad concepts of terrain and moving objects. Thus, customized versions can be created based on different editing rules. For example, the DATS editor takes a simple approach to terrain, assuming the world to be flat and featureless. This is adequate for naval air problems. A different version could implement a more sophisticated terrain model and be suitable for land forces scenarios. A yet more sophisticated version could accommodate the visibility and routing capabilities needed for urban scenarios.

The scenario editor has an interface that will be familiar to Windows users. As such, it will be easy to learn and use. It is implemented using the new Microsoft .NET technology and is thus compatible with their new and upcoming desktop operating systems.

Initial Screen Layout

The DATS Scenario Editor (DSE) is designed to give users a consistent, easy to manage, interface. It combines a basic fixed layout with pop-up windows that can be invoked, arranged, and deleted as the need arises. The main window displays menus and tools in the conventional locations. A status bar at the bottom displays information depending on the current context and user action.

[1] Grushcow, M., "A Task to Investigate Information Usage in Air Defence Decision Support Systems", August 2003, Contractor's Report, Contract W7711-0-7687-21

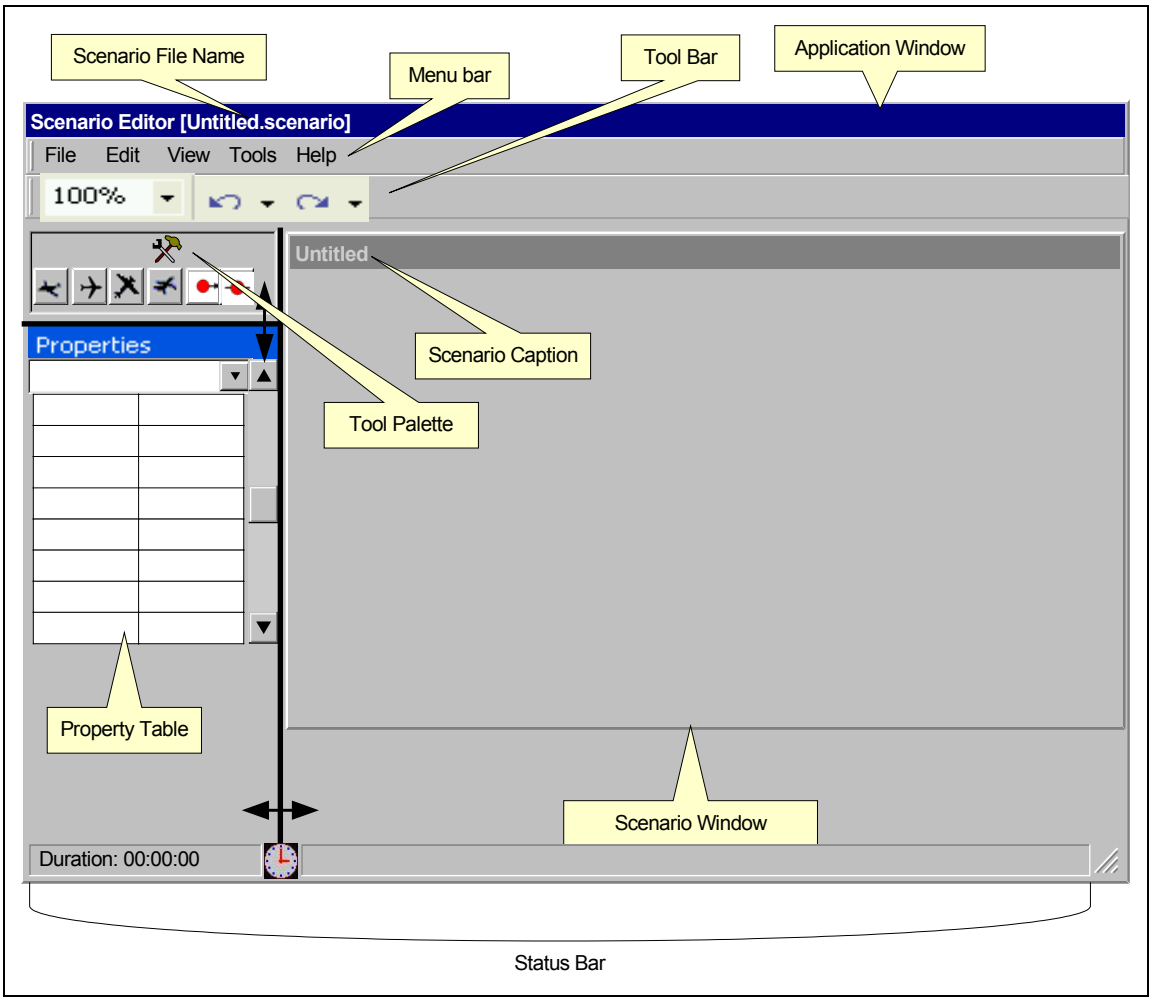


Figure 1. *Initial Screen Layout*

The user has control over a vertical splitter that separates the left and right hand space and the horizontal splitter that separates the toolbox from the property table. Reducing a panel to below a certain size will cause scroll bars to appear so that the user can access all of the contained information. The toolbox contains the various types of controls that can be placed in the scenario window. The properties window displays the properties of the currently selected object in the scenario. The contents of the status bar varies, but always includes at least the scenario *mode indicator* and duration. The duration is the total elapsed time of the scenario, from start to the exit of the last track. The mode indicator is described below (Section: Managing Time, Speed, and Distance).

Menu Structure

The DATS menu structure follows the standard Windows format.

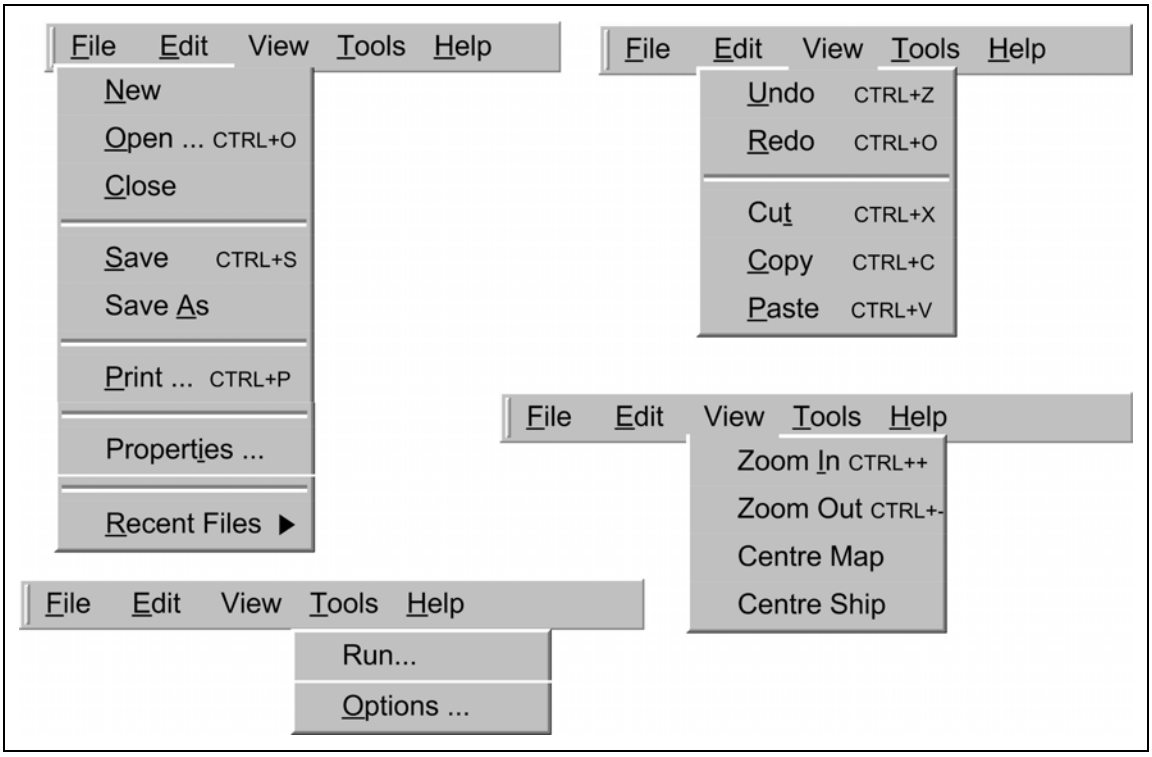


Figure 2. *DATS Menu Structure*

Menu options are enabled or disabled based on whether or not they are applicable to the current context.

The Toolbar

The toolbar also follows standard Windows conventions. The zoom tool controls scaling of the scenario window. The other tools invoke undo and redo functions. The number and types of undo actions will be determined during design and implementation stages.

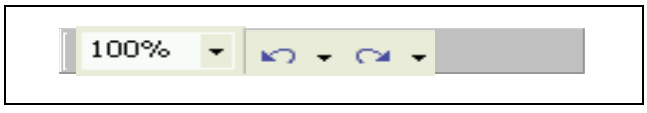


Figure 3. *DATS Toolbar*

Other tools may be added to the toolbar in response to user feedback.

The Toolbox

The toolbox contains items that can be placed in on the scenario map. Typically the items on the box represent aircraft and route markers. More generally, these tools simply represent objects that carry sets of attributes. The specific set of tools depends on external configuration information. This allows the scenario editor to be used for a variety of different purposes. For example, a set of tools could represent different types of units involved in a ground exercise.

Figure 4. *The Toolbox*

Users drag tools from the toolbox to the scenario window. This is described in detail below (Section: Placing Targets).

The Properties Window

The properties window displays the properties of the currently selected object. The drop down control below the title shows a list of all of the objects in the scenario. Selecting an object from the list *makes it the selected object* and displays its properties.

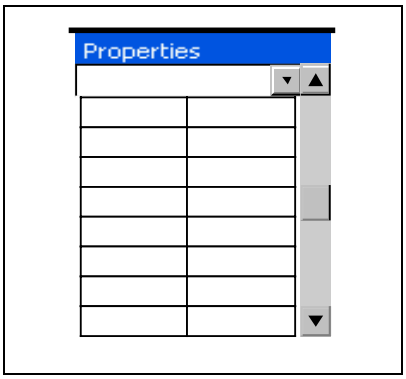


Figure 5. *The Properties Window*

Properties are listed alphabetically with the property name on the left and property value on the right. Properties may be settable or only displayable (read only). Read only property names are *greyed out*. For example, *wings dirty* might be a fixed property with the value *true* for certain types of controls. Another type might also have a *wings dirty* property with the fixed value *false*. *Wings dirty* might be a settable property for some other type of control.

All objects that can be placed in the scenario (with the exception of the map itself) do share a minimal set of properties: location, altitude, direction (course) and speed.

As described in the DATS functional specification, properties may be derived and/or interrelated. For example, setting direction will determine CPA (closest point of approach). Setting the departure speed at one waypoint will determine the arrival time at the following one.

Creating a New Scenario

Clicking on the scenario window makes the scenario itself the selected object. This allows the author to specify its size and geographic bounds. Clicking on the ellipsis on the *Map* attribute will bring up a standard Windows file open dialog. The author navigates to the image they wish to use and then clicks the *OPEN* button. Note that the image itself becomes included in the scenario. The file reference that appears as an attribute is simply an annotation. If that file is later deleted or changed, the scenario will not be modified in any way.

Note also that no objects can be moved from the toolbox to the scenario window until the size and bounds have been specified. Similarly, the size and bounds may not be changed if there are any objects in the window.

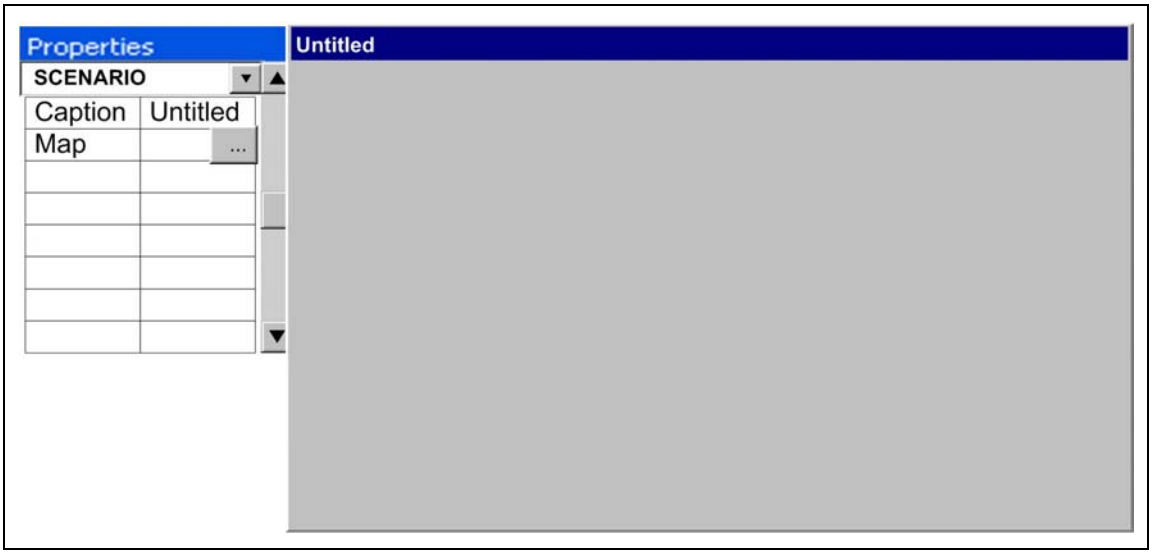


Figure 6. *Empty Scenario Window with Property Sheet*

Once a map has been selected and the other information provided, the scenario window will look as follows:

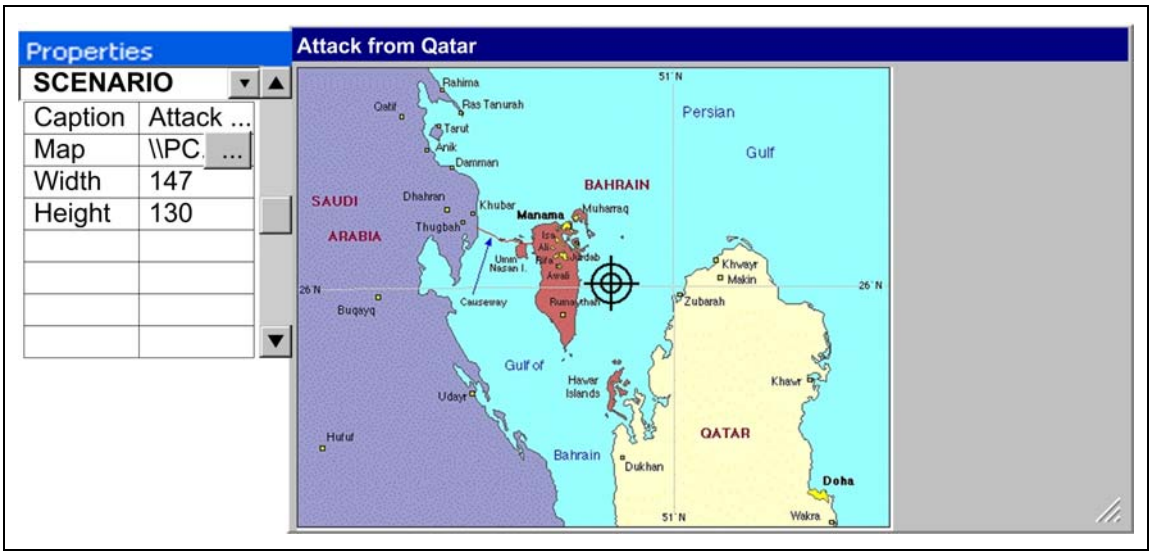


Figure 7. *Initialized Scenario Window*

The scenario window can be resized, although the map will always retain its "natural" aspect ratio. Ownship is initially placed in the centre of the map, although it can be dragged around the window, within the map borders.

Once the map dimensions have been set, the information will be added to the status bar at the bottom of the main window.

Figure 8. *Map Status Information*

The size information represents the width and height in nautical miles.

Manipulating Ownship

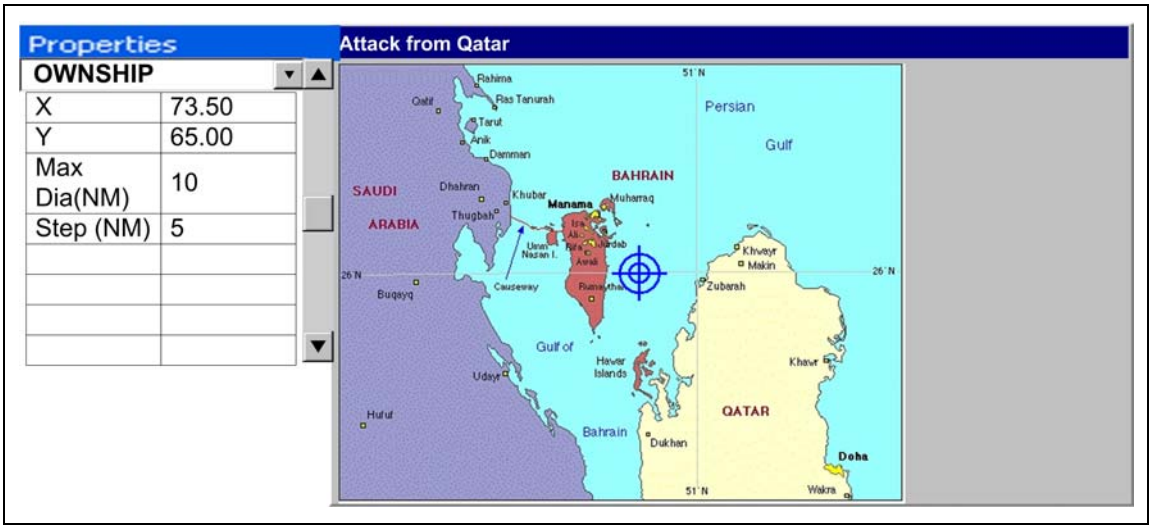


Figure 9. *Ownship Properties*

Ownship can be moved by dragging it within the bounds of the map. Alternately, setting X and Y (relative to the bottom left map corner) in the property sheet will cause it to be placed at that precise location. Concentric rings will be placed around Ownship in steps of *Step* NM., until a diameter of *Max Dia*. is reached. These rings assist in mission planning and the properties can be changed at any time. Note however that they are saved with the scenario and will be used in representing Ownship when the task is run.

Note also that some route planning features work relative to Ownship's location. Routes set up in this way *will not* be automatically adjusted if Ownship is moved afterwards.

Zoom, Pan, and Scroll

Zooming is controlled by the Zoom In and Zoom Out menu items and the zoom combo box on the toolbar. Regardless of which method is used, the zoom control on the tool bar will reflect the current scale factor. Zooming attempts to scale around the current centre of the scenario window. The circles around Ownship will scale, but other items such as targets and waypoints will not.

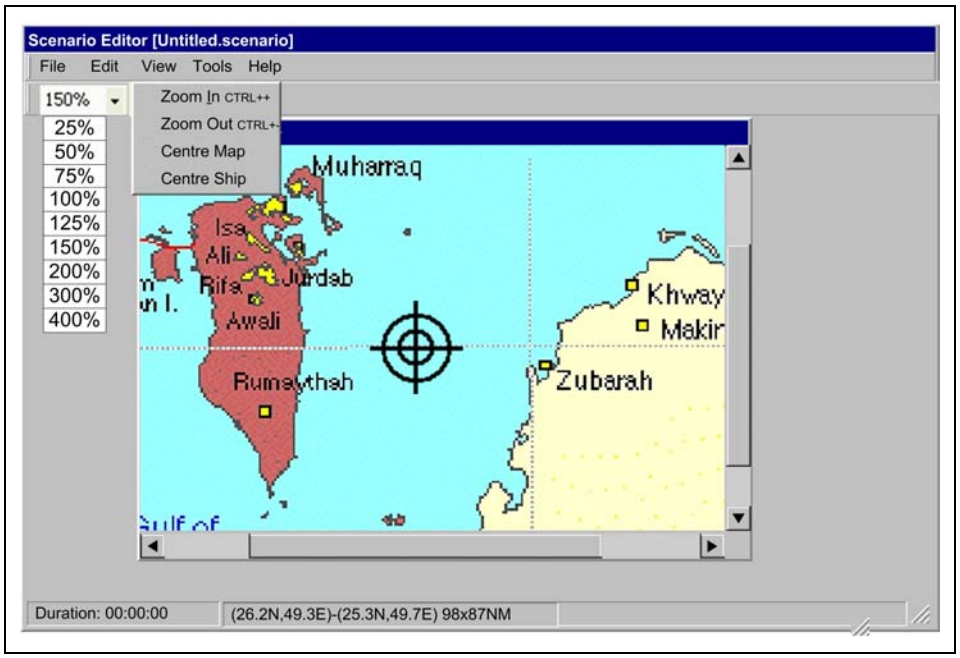


Figure 10. *Zoom, Pan, & Scroll*

Note that since the user can control the size and shape of the scenario window, its aspect may not match that of the map. This means that in some cases, the map will occupy less space than the window itself. Magnification factors are always applied relative to the map space, not the window size. Depending on the graphics format used, the map may become increasingly coarse as it is magnified and may lose information as it is shrunk.

The Centre Map menu item will scroll and pan the map so that it is centred at the current scale. The Centre Ship menu item will scroll and pan the map so that Ownship is as close as possible to the centre of the scenario window at the current scale.

The map status information in the main window's status bar will reflect the map area displayed in the scenario window.

Placing Targets

Targets are placed in the scenario window by selecting dragging them from the toolbox to the desired location.

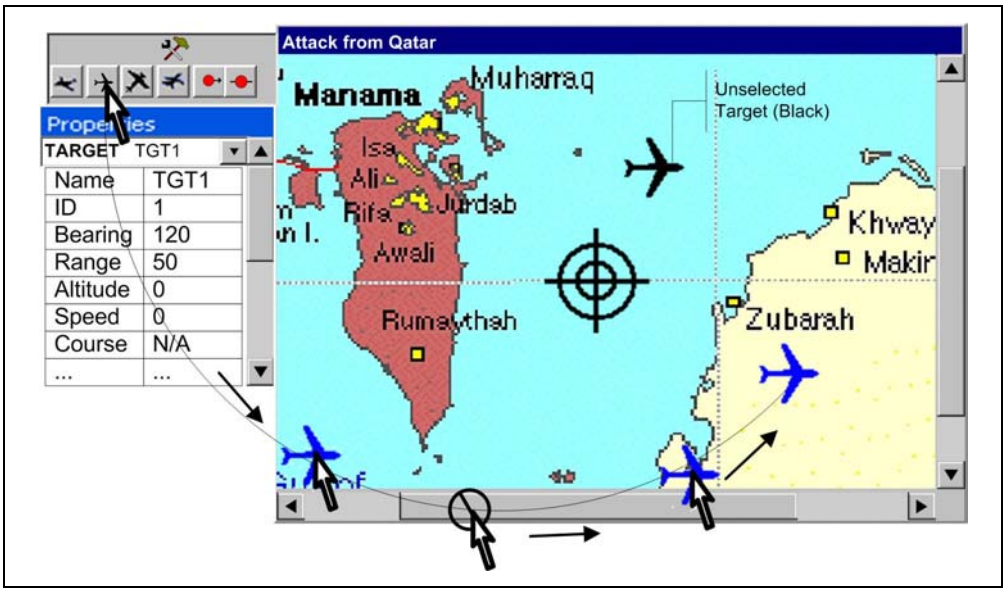


Figure 11. *Placing Targets*

When the mouse is not over the map area, the forbidden icon (⃠) is displayed. If the mouse is released at this time, the drag is cancelled.

When the target is dropped on the map, all of its properties appear in the properties window with appropriate initial values. Bearing and range are relative to Ownship.

A target can be moved by dragging it over the map. The same rules as above apply when the mouse is not over the map. It can also be moved my modifying its bearing, range, latitude and longitude properties. These values are also constrained by the map dimensions (regardless of the current scaling).

Creating Routes

A route consists of a number of flight segments (legs) delimited by *waypoints*. The route begins at the target, which is otherwise considered equivalent to a waypoint. Waypoints and legs have the following characteristics:

1) The target's course and its rate of ascent or descent are derived from the relative location of the waypoints.

2) A target's properties are set each time it begins a leg, based on the properties of the entry waypoint.

3) Once established at the entry waypoint, a target's properties are fixed for the duration of the leg.

4) Each waypoint has an arrival time. This is the same as its departure time. There is no loitering.

5) The arrival time associated with the target is the time at which it enters the scenario and the arrival time associated with the last waypoint in a route is the time at which the target is removed from the scenario.

6) The duration of the scenario is determined by the latest waypoint arrival time and is shown on the main window status bar.

7) The relationship between time and space is described in detail in the "Route Planning" section below.

Creating the Initial Route

A user can start a route by clicking on a target with the right mouse button. This will display the Add Waypoint context menu.

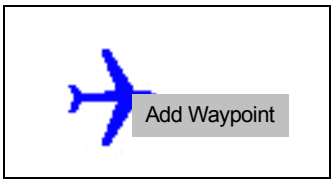


Figure 12. *Starting a Route*

At this point, the cursor will change into the Route Marking Cursor, shown below.

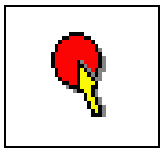


Figure 13. *Route Marking Cursor*

The process of laying waypoints is shown in the following figure. The Route Marking Cursor is "sticky," that is, once it is shown, the user does not have to keep the mouse button depressed. Every click of the mouse drops a waypoint. Rubber banding is used so that the path from the previous waypoint to the cursor is shown explicitly. The cursor is returned to its normal mode by dropping a waypoint with a double click, by depressing the ESCAPE key or by selecting any other tool. Note that the cursor is temporarily returned to the normal mode whenever it is not over the map area. This allows the user to access the scrollbars.

Any time a waypoint is dropped, it becomes the selected object and its properties are displayed in the property window. This allows users to drop a waypoint, configure its properties and return to the map to drop another without having to change tools. Changing the waypoint's location properties will reposition it in the scenario window.

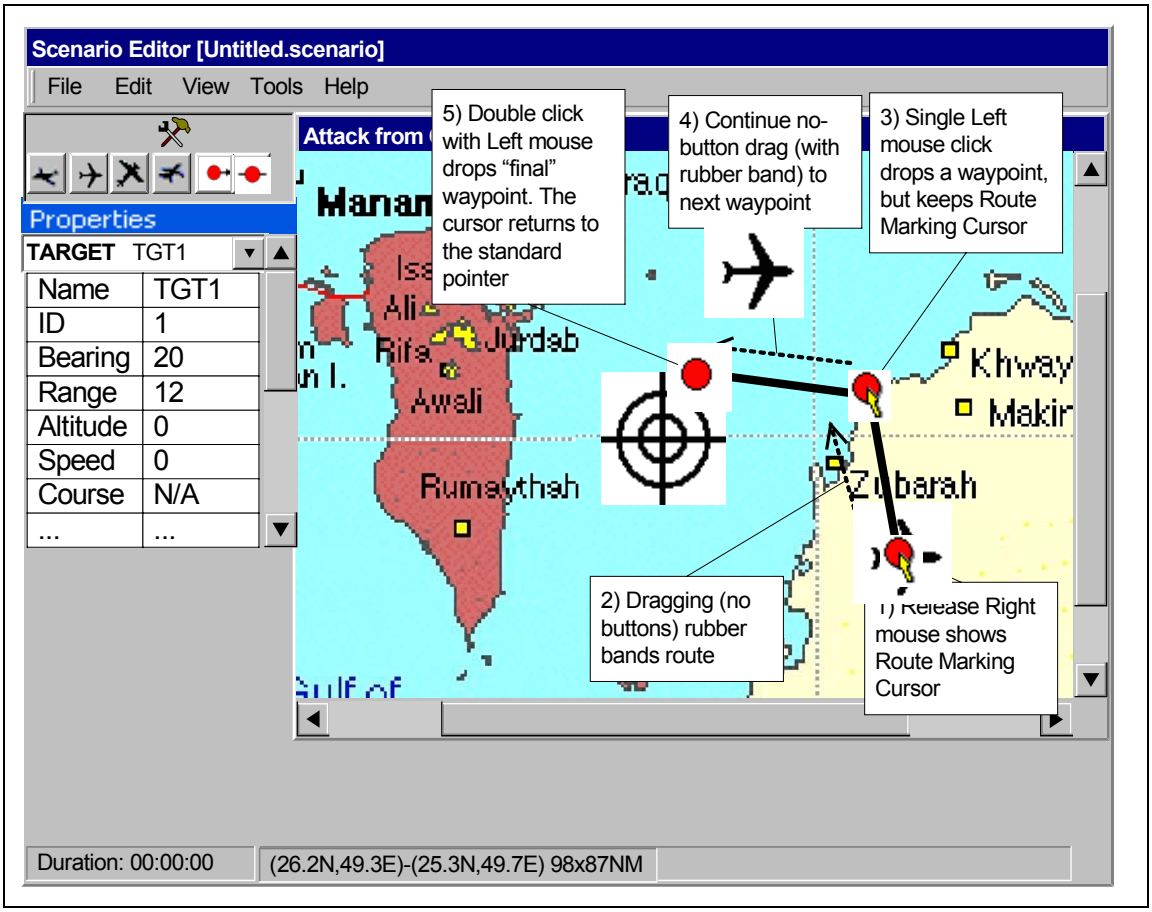


Figure 14. *Laying Waypoints*

Deleting, Moving and Inserting Waypoints

A waypoint can be deleted by selecting it and depressing the DELETE key. If the waypoint is an intermediate one, it and the adjacent legs are removed and the preceding and following waypoints are joined with a single leg. If it is the terminal waypoint in a route, it and the preceding leg are deleted and the preceding waypoint becomes terminal. If the target itself is deleted, it and its complete route are removed from the scenario.

A waypoint can be moved either by dragging it with the mouse or selecting it and modifying any of its location properties.

The process of inserting a waypoint is shown in the following figure.

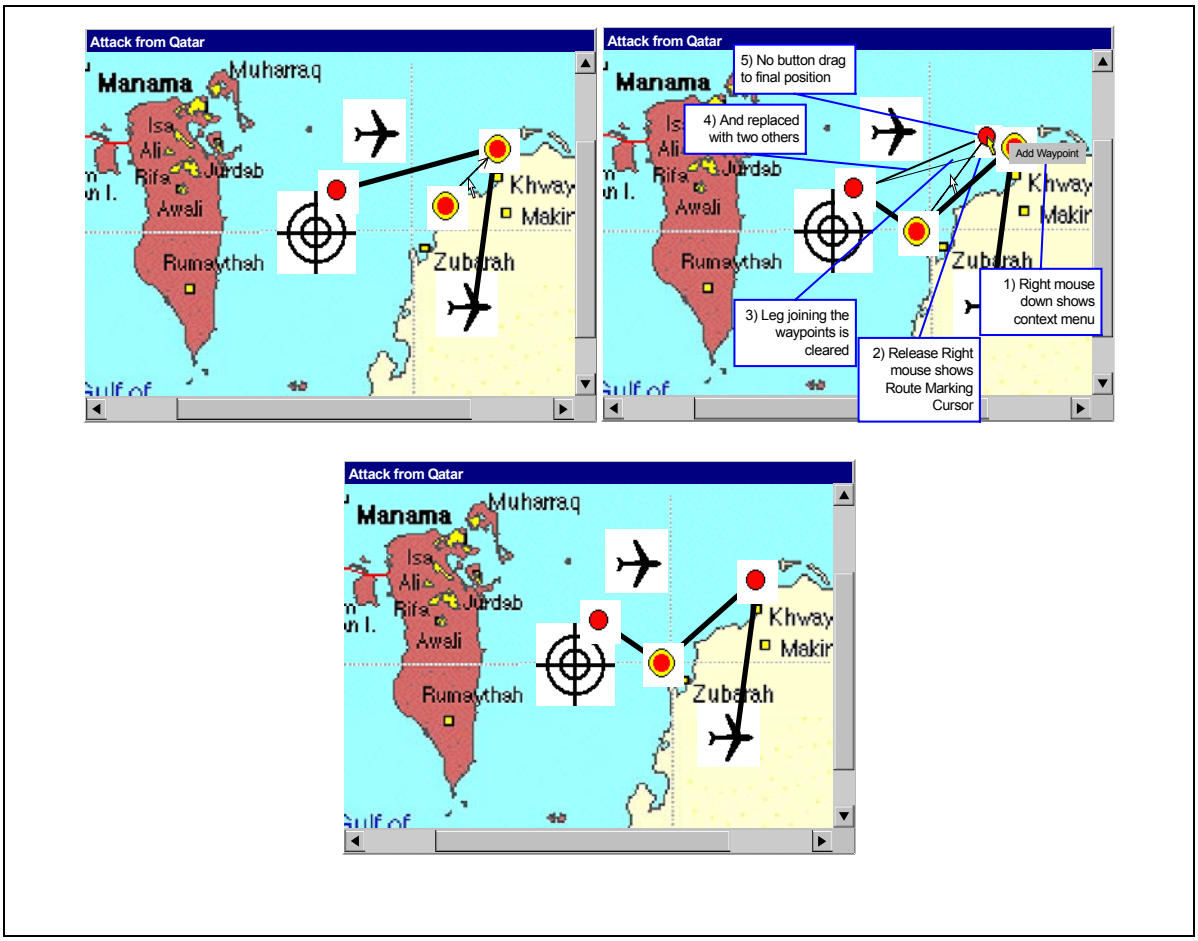


Figure 15. *Inserting Waypoints – Before, During, and After*

The new waypoint is always added after the selected one. The rules and procedures for inserting waypoints follow those for adding initial waypoints as described in "Creating the Initial Route," above. The only difference is that the inserted waypoint is always connected to the one that follows the insertion.

Managing Time, Speed, and Distance

Creating a "flight plan" for a target involves speed and time as well as distance. These attributes are obviously related and changing any of them has implications for the others. The previous sections have dealt with route planning; we are now concerned with when the target will reach each waypoint and how it will traverse the intervening legs. We will be manipulating the following concepts:

Location
A waypoint has a location that can be set by dragging it to a position on the map or setting its bearing and range attributes.

Distance
The distance between two adjacent waypoints is calculated from their respective coordinates and altitudes, assuming that the world is flat.

Time
The time associated with a waypoint or initial target location is the time at which the target reaches the location. For initial target location, this is the aircraft's entry time in the scenario and the time when it becomes visible. The time of the last waypoint is the time the target disappears. For intermediate waypoints, the time is the arrival and departure time of the aircraft. Aircraft do not tarry.

Course
"Course" is a read-only property. It is defined as the direction to be traveled from one waypoint to the next waypoint.

Speed
An aircraft moves at fixed speed between waypoints. Each time it arrives at a waypoint, its speed and course change instantaneously to the values specified for the waypoint.

Since *Distance = Speed * Time*, moving a waypoint will have consequences for at least the adjacent legs. The user may wish to hold either speed or time constant, depending on how they are planning the scenario. To accommodate this, the scenario editor will operate in either "Hold Speed" or "Hold Time" mode. The mode will be indicated by an icon in the status bar at the bottom of the main window.

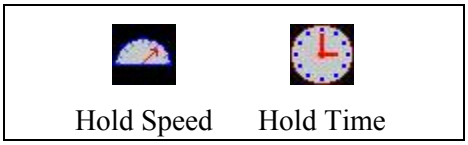


Figure 16. *Speed and Time Modes*

Hold Time will be the initial mode. Clicking on the icon will cause it, and the editor, to flip between modes.

In Hold Speed mode, moving a waypoint will cause the arrival time for that waypoint and all subsequent waypoints to be updated. Moving the initial position of the target will only affect subsequent waypoints. That is, the time of appearance of the target in the scenario will not be changed.

In Hold Time mode, moving a waypoint or initial position of the target will only affect the departure speed of that waypoint and the previous one, if any.

Planning for Closest Point of Approach

CPA is considered an important track characteristic when determining target priority. It is determined from the target's course and the relative positions of the target and Ownship. Note that for DATS, CPA is constant between waypoints. CPA is a target (and waypoint) characteristic like course, speed, arrival time, bearing and range, and like these, it can be specified by the user in the editor. Of course these characteristics are interrelated and changing one has impact on others. This section explains what happens when specifying CPA for a leg modifies a flight plan.

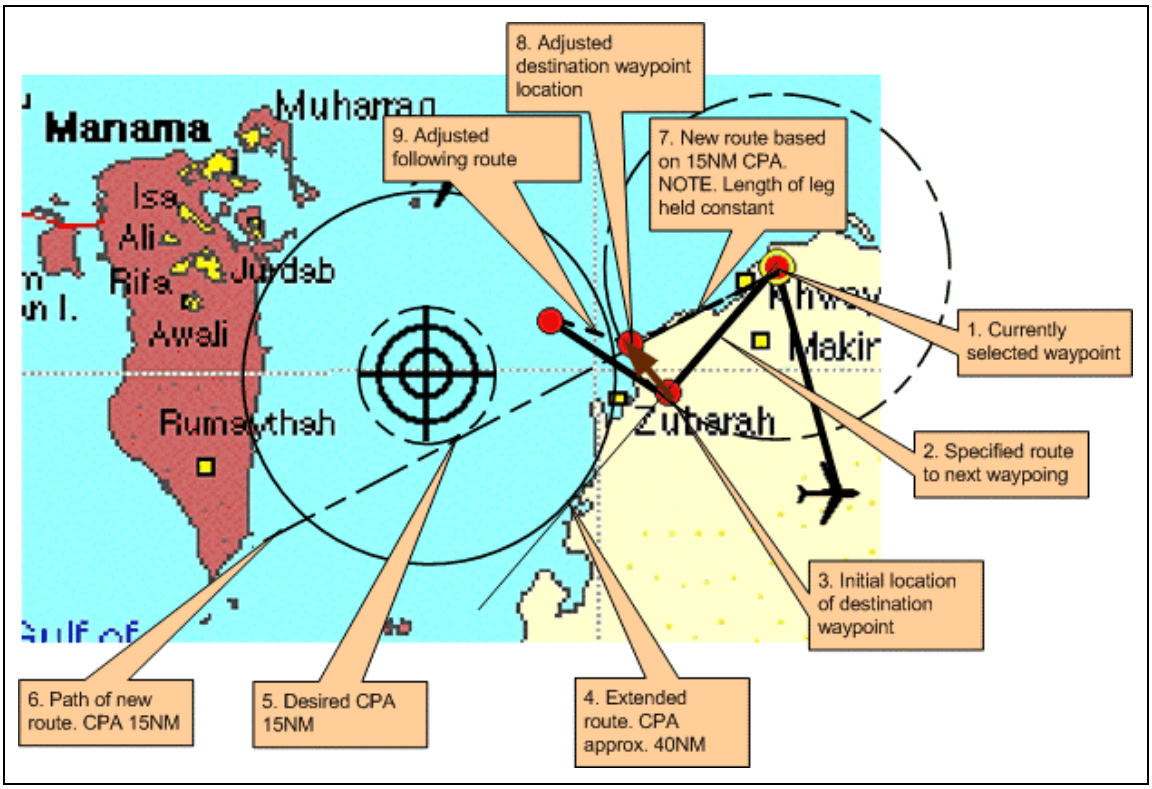


Figure 17. *CPA Planning*

The following points are numbered corresponding to the above.

1. Select a waypoint. Its properties will be displayed. These include course, speed, and CPA.
2. These properties determine the path of the outbound leg.
3. That leg terminates at a destination waypoint.
4. If we extend that path, the CPA would be 40 NM.
5. We specify a CPA of 15 NM. The dashed circle around Ownship shows this distance.

6. The corresponding course from the selected waypoint is shown as a dashed line.
7. The new route follows the newly determined course.
8. We have to move the destination waypoint. We do this by following the new course and keeping the distance fixed. The dashed circle around the selected waypoint shows this.
9. Note that we must now adjust outbound course at the destination. Since the length of the leg to the following waypoint, outbound speed or terminal arrival time must be adjusted. Which on depends on whether the editor is in “Hold Time” or “Hold Speed” mode.

Displaying and Modifying Flight Plans

Right clicking on a target that has a route will display a context menu that includes a “Flight Plan” item. Only one window per flight plan will be displayed, but windows for multiple flight plans can be displayed at the same time allowing for comparison and coordination.

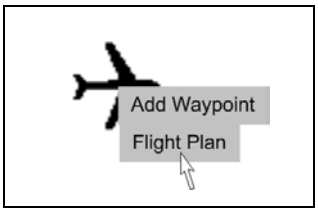
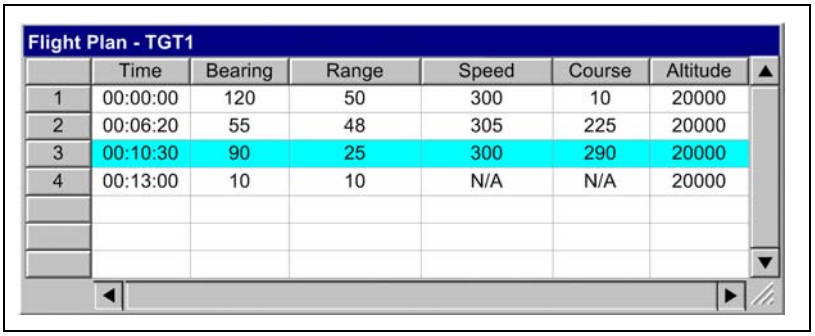


Figure 18. *Requesting Flight Plan Information*

Information is displayed in a grid as shown in the following figure:



	Time	Bearing	Range	Speed	Course	Altitude
1	00:00:00	120	50	300	10	20000
2	00:06:20	55	48	305	225	20000
3	00:10:30	90	25	300	290	20000
4	00:13:00	10	10	N/A	N/A	20000

Figure 19. *Flight Plan Information Grid*

The grid has the following behaviour and interactions with other components:

- Columns can be reordered by dragging.
- Columns can be deleted by selecting the column header and hitting the DEL key
- Columns can be added. The choice of columns is a function of the target type and any characteristic of the selected target type can be added to the grid. Note that not all target types have the same set of characteristics.

- Clicking on a cell selects (highlights) the row and makes the corresponding target or waypoint the selected object.
- Left clicking on a target or waypoint (making it the selected object) will highlight the corresponding row in the flight plan window for the target *if* it is currently displayed.
- Clicking on a cell allows the its contents to be edited (if this is permitted). Changing a value in a cell will change the corresponding characteristic for the target or waypoint.
- Changes to values in cells will be reflected in the properties window if the corresponding target or waypoint characteristics are currently being displayed there.
- Similarly, changes made in the properties window will be reflected in the flight plan window if it is currently displayed.
- If the target or waypoint is moved or deleted by operating on the map window, the corresponding changes will be made to the flight plan and properties windows if they are displayed.

Scenario Properties

The scenario itself, like a Word document, has a set of properties. These are can be accessed through the File|Properties menu item. Scenario properties are presented in "tabbed notebook" format. The first page shows general properties and these are read-only.

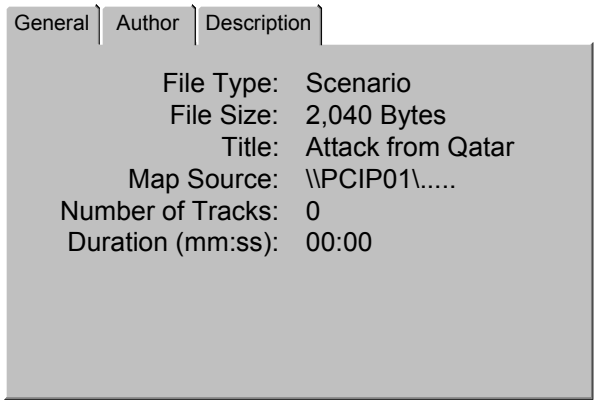


Figure 20. *General Scenario Properties*

The second page deals with authorship. This identifies the people who designed the scenario and those who actually implemented it with the editor.

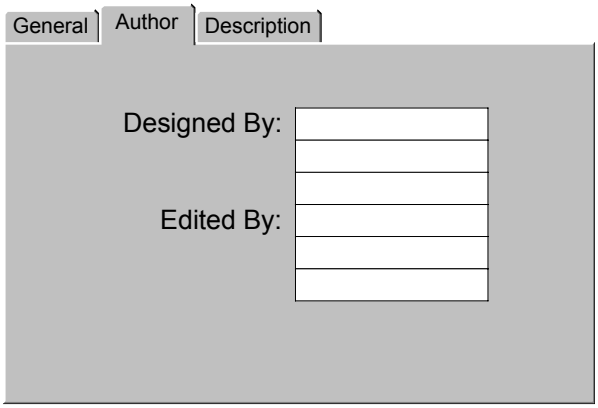


Figure 21. *Authorship Properties*

The final page provides general geographical and historical context information.

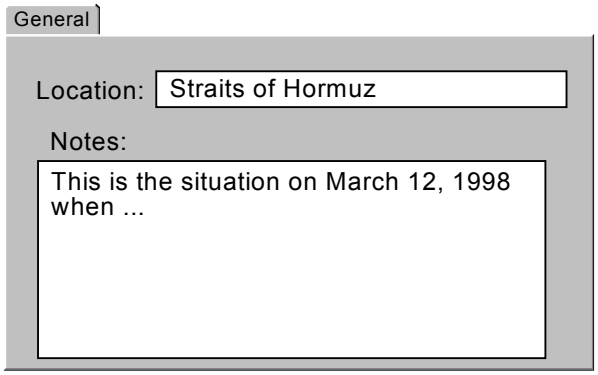


Figure 22. *General Scenario Properties*

Simulating the Scenario

The author can preview the scenario at any point in the editing process. Selecting the Tools|Run… menu item will display the following window:

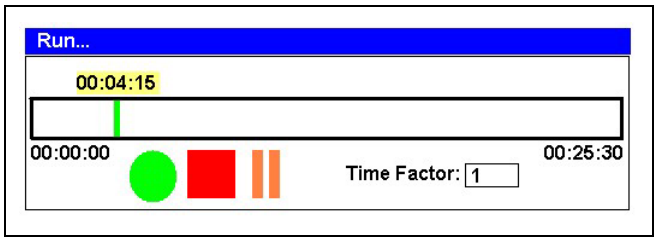


Figure 23. *The Simulation Control*

The vertical green bar (current time indicator) will initially be at the left edge of the timeline. The scenario duration will be indicated at the right end. The scenario will play at real time times the time factor entered in the "Time Factor" text box. This may be fractional for slow motion.

The current time indicator can be dragged to any point on the time line. Clicking the green "Run" button will start the scenario from that point. Hitting the orange "Pause" button will stop the simulation at its current point. The red "Stop" button will reposition current time to the beginning.

When the run menu item is selected and the window displayed, all targets and routes will be cleared from the display. Targets will then be displayed according to their location at time zero. Dragging the current time indicator to a new time and then releasing it will cause the target positions to be displayed based on the new time. Note that targets that enter after, or leave before the selected time will not be visible.

Clicking on the Run button will cause the scenario to run forward from the current time at the specified rate. When the scenario is running, the scenario window cannot be resized, zoomed, panned or scrolled. The scenario will run until it hits the end time or the user clicks on the Pause or Stop buttons.

Any time the scenario is not running, the user can select targets. These may not be modified, but their properties will be displayed in the property window.

General Considerations and Features

About Properties

Each type of object in the editor has a set of properties that the user can manipulate through the property window. In general, these fall into three broad classes; Class 1 properties are of

direct concern to all scenario editors, i.e., *the framework*, Class 2 properties are of concern to a particular type of editor, and all others are Class 3 properties. For example, all of the properties associated with the scenario window itself (map, lat., long., etc.,) fall into the first class since all editors require them in order to operate. Similarly, target properties like course, speed, etc., also fall into the first class. Other properties such as bearing and range fall into the second class since they are derive from the concepts of naval air defence. Finally, properties, such as "wings dirty" for a target fall into the third class. They are of interest to the author, but the editor itself is simply responsible for keeping track of them and ensuring that they are validated.

Taking this approach to properties makes the scenario editor very generic, and DATS can be considered to be simply its first client. DATS targets exemplify the broader class of *moving objects* that might include land forces, individuals, naval craft, and so on. The DATS editor is very simple and considers the world to be flat and featureless. This could be extended in another editor designed to deal with urban environments where routing and visibility would be highly constrained.

The models that underlie properties and their containers have to support the intent to develop a generic editing system. This has several implications including:

1) The toolbox itself has to be populated at runtime based on external tool specifications. These would include related icons (e.g., selected and unselected), and (class 2) properties.
2) The properties will fall into various classes based on the type of value that they can hold, e.g., unrestricted strings, constrained numeric, choices, etc. Each property will include an indicator of its editing rule.
3) (2) implies that the property window has to be constructed dynamically and populated based on the current object. This allows for different types of targets with similar but different properties.
4) We will accept the restriction that all Class 3 properties can be validated independently of any other Class 3 property.
5) As described in the original functional specification, some properties are *derived*. CPA is an example. Derived properties are calculated either by the specific editor (DATS) or the framework. As with editing rules, each derived property will include an indicator of its property value calculator.

User Interface Behaviour

The scenario editor is intended to operate as a standard Windows application. As such, it will use standard Microsoft interaction mechanisms.

Undo/Redo will be implemented to the extent allowed by time and cost.

Scenario authoring activities largely consist of placing objects and setting their properties. The scenario window will visually highlight the “current object” in a way that makes it easily visible. If the current object is deleted, the system will select and highlight another object as the new current object if a natural successor can be identified. If not, there will be no current object.