RL-TR-91-239, Vol I (of two)
Final Technical Report
September 1991

GRAPHICAL REQUIREMENTS FOR FORCE LEVEL PLANNING

General Electric Company

George D. Montanaro, William J. Schroeder, Boris Yamrom, William Lorensen, Dipen Moitra, Peter M. Meenan

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-239, Vol I (of two), has been reviewed and is approved for publication.

APPROVED:

EARL C. LABATT, JR.
Project Manager

FOR THE COMMANDER:

RAYMOND P. URTZ, JR.
Technical Director
Command, Control, & Communications Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL(C3AA) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1991	Final Jun 89 - Mar 91

4. TITLE AND SUBTITLE

GRAPHICAL REQUIREMENTS FOR FORCE LEVEL PLANNING

5. FUNDING NUMBERS

C - F30602-89-C-0078
PE - 62702F
PR - 5581
TA - 24
WU - 32

6. AUTHOR(S)

George D. Montanaro, William J. Schroeder, Boris Yamrom, William Lorensen, Dipen Moitra, Peter M. Meenan

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

General Electric Company
P.O. Box 8
Schenectady NY 12301

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

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

Rome Laboratory (C3AA)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

RL-TR-91-239, Vol I (of two)

11. SUPPLEMENTARY NOTES

RL Project Engineer: Earl C. LaBatt, Jr./RL/C3AA/(315)330-7764.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report presents the findings of the analysis phase of the contract, Graphics Requirements for Air Force Tactical Decision Aids. The purposes of this report are to describe the graphical needs or requirements associated with Air Force tactical planning, and to describe 2D and 3D graphical techniques that address these requirements. Focus is placed on the various planning tasks. Maps, threat zones, weather, and route planning were found to be pervasive throughout all the planning phases, and are addressed in sections dedicated to those topics.

14. SUBJECT TERMS

Graphics, Tactical Planning, 2D Graphics, 3D Graphics

15. NUMBER OF PAGES

[illegible]

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	SAR

NSN 7540-01-280-5500

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

List of Figures vi

1 Introduction **1**
- 1.1 Traditional Planning Approaches 1
- 1.2 Graphical Techniques for Force Level Planning 1
- 1.3 Rapid Prototyping Using LYMB 2

2 Selecting Targets, Bases and Weapons **4**

3 Mission Scheduling **9**
- 3.1 Representing Sorties 9
- 3.2 Tactical Constraints 11
- 3.4 Resource Availability 13

4 Resource Allocation Networks **16**
- 4.1 Traditional Techniques 16
- 4.2 Defining Resource Allocation Networks 17
- 4.3 Managing Conflicting Resource Needs 17
- 4.4 Risk Reward Analysis 20
- 4.5 Impact of Weather on Planned Missions 20

5 Coordinating Missions **23**
- 5.1 Constraint-Based Mission Planning 23
- 5.2 Using a Constraint Based Planner 24

6 Strategic Situation Assessment **28**
- 6.1 Basic Techniques 32
- 6.2 Multi-Valued Sitmaps 35
- 6.3 Animated Sitmaps 35

7 Conclusions **36**

8 References **38**

List of Figures

Figure 2.1 Targets, Bases and Mission Windows 5
Figure 2.2 Target, Query & Feasible Weapons & Aircraft 6
Figure 2.3 Resource Allocation and Mission Scheduling 8
Figure 3.1 Example TOT Slider 10
Figure 3.2 Two TOT Sliders Linked by a Time Constraint 12
Figure 3.3 Resource Plot Depicting Resource Allocation 14
Figure 4.1 Resource Allocation Network 18
Figure 4.2 Symbology for Resource Allocation Network 19
Figure 4.3 Analyzing Cost, Benefit & Risk Tradeoffs 21
Figure 5.1 Coordinating Refuelling with Two-Step Bombing 25
Figure 5.2 Mission Coordination Animation 27
Figure 6.1 User Interface to Sitmap Simulation 30
Figure 6.2 Sitmaps for Initial Configuration in Figure 6.1 31
Figure 6.3 Sitmaps After Moving Time Slider 33
Figure 6.4 Sitmaps After Moving Fourth TOT Slider 34

1 Introduction

The scope of the **Graphics Systems Representation** project is to develop novel graphical techniques for performing Air Force planning tasks. The primary nature of these planning tasks relate to: deploying feasible combinations of aircraft and weapons against a given set of targets, determining the availability of resources at various bases over the course of the planning interval, performing a cost benefit tradeoff among alternative uses of the given resources, coordinating inter–dependent missions, and assessing the strategic value of the available and deployed resources and also the strengths and weaknesses of enemy forces.

1.1 Traditional Planning Approaches

Traditional techniques for these planning tasks are primarily text based, and most user interaction is based on tabular forms. For example, the user can type in the name of a target, and the system will display a list of aircraft and weapons which can be deployed against that target, together with a list of the bases at which those resources are available. Often, these tools are used in conjunction with an expert system, which can make a feasible allocation of resources to targets and also tries to maximize the potential value of deploying the resources.

What is lacking from these traditional approaches is the ability to visualize the resource allocation tasks in relation to the geography of the gaming area, as well as the ability to visualize the inter–dependence between the various planned missions. Hence it becomes difficult for the user to see why particular resource deployment decisions were made or what would be the effect of modifying a selected subset of the mission parameters. Also lacking is the ability to visualize the costs, risks, and benefits associated with the various feasible options for deploying the available resources against alternative targets.

1.2 Graphical Techniques for Force Level Planning

In the ensuing sections, we will describe some novel graphical techniques which we have devised for visualizing the planning alternatives and their relationships to each other and also for interactively specifying the planning constraints and modifying the decision variables. Section 2 describes how to view and select targets in relation to the geography of the gaming area and how to select weapon and aircraft combinations which may be feasibly deployed against those targets. Section 3 discusses techniques for scheduling missions, determining the availability of resources at the various bases during the planning time frame, and for managing tactical precedence constraints among different missions. Section 4 describes techniques for visualizing the resource allocation alternatives and for analyzing the cost benefit tradeoffs. Section 5 describes how to represent the geographic and temporal constraints among a set of missions and how to derive a coordinated set of mission plans which obeys the given constraints. Finally, Section 6 describes how to visualize the strategic and tactical value associated with a given deployment of resources, the geographic disposition of enemy targets and offensive capabilities, and the temporal and spatial effects of executing planned missions or moving friendly assets.

The main strength of our techniques stems from the ease with which the user may visualize all the facets of the given planning tasks. Our techniques provide greatest support during the following steps of the planning process:

- Visualizing the geographical relationships among the targets and the assets.
- Visualizing all the available resource allocation alternatives.
- Determining all the tactical, temporal, and resource directed dependencies among the set of planned missions.
- Determining the cost benefit tradeoffs among the various resource allocation alternatives.
- Suppressing the display of all information which is not relevant to the specific subtask on hand.
- Assessing the strategic and tactical impact of the planned missions on the spatial and temporal distribution of offensive and defensive capabilities.

1.3 Rapid Prototyping Using LYMB

We used an object-based software development system called **LYMB** (developed at GE CRD), to create graphical prototypes of our planning techniques. Short example LYMB scripts are used in this document to illustrate the relative ease with which one can manipulate the various objects we have created. In this section, we give a brief overview of LYMB, so the reader unfamiliar with it can understand the examples.

A LYMB *object* consists of some data and a set of operators that manipulate the data. The sole interface to an object's data is through the operators. *Classes* are template objects used to create *instances* of a class. Following Smalltalk parlance, in LYMB, an object's data is referred to collectively as its *instance variables*, while its operators are called the *messages* to which it responds.

LYMB is a dual-language system. New classes are defined by writing C code, while instances are created and manipulated by an interpreted scripting language. In this document, all examples are written in the LYMB script language. By default, when LYMB is run, it prompts for script input on the terminal.

LYMB's script language is syntactically very simple. Each statement names an object and one or more messages to send to that object. A LYMB statement is terminated by a semicolon (";"). To create a new floating point number simply type "scalar new: s;", and a carriage return to the LYMB prompt. In this case, the object is "scalar", and the message is "new:". The next token on the line, in this case "s", is an argument for the message. The semicolon terminates the statement.

LYMB provides a convenient shorthand notation. By default, the object that receives the first message in a statement will receive successive messages in the statement. Having created s, we can send it a series of messages in a single statement while only naming it once[1]:

```
scalar new: s
      = 10                   -- set its value to 10
      * 5                    -- then multiply it by 5
      + 4;                   -- finally, add 4
```

Note that a single LYMB script statement can cross lines. (The multi-line format shown above is the normal one used in this document for LYMB statements.) In the example above, s is first sent an " = " message, with an argument of 10. It is then sent a "*" message with an argument of 5. Finally, it is sent a " + " message with an argument of 4. After this chain of messages, the value of s is 54 (10 * 5 + 4). Comments can either be C-style (text enclosed in "/*", "*/" pairs), or Ada-style (text following "—", extending to the end of the line).

This should provide the reader with enough information to read the simple LYMB scripts that appear in this document. More detailed information regarding LYMB can be found in [1, 2].

1. The major exception to this rule is the "new:" message which establishes the newly created instance as the recipient of successive messages in the statement, not the object to which the new: message was sent (which is usually a class).

2 Selecting Targets, Bases and Weapons

The first task during mission planning involves visualizing the spatial relationships among the targets and the bases which are within range of those targets, as well as the relationship of the targets and bases to the geography of the gaming area. Traditional text oriented techniques for accessing this information deny the military planner the most powerful and time-honored planning tool: the wall map with colored flags and pencilled paths.

A map of a user-selected portion of the *gaming area* is presented in a workstation window. A planner can choose a target by *pointing and clicking* with a mouse. Once a target is chosen, the map is modified (or another window is created) to display only the area of concern for that target. Zooming into or panning across the area covered by the map triggers *database queries* on the underlying database of the gaming area, to extract all the information that pertains to the objects currently displayed on the map. Using a menu driven interface and mouse buttons, the planner can look at any desired information related to the objects visible on the current map. (See Figure 2.1.)

The information for targets falls into 4 categories:

- Data on the physical composition and layout of the target.
- Combinations of weapons and aircraft that can feasibly eliminate the target and the associated kill probabilities.
- Bases which are within range of the target and have the appropriate weapons and aircraft available.
- Information about the planned missions which relate to the targets visible on the map.

The physical data relating to the targets and the feasible weapons and aircraft combinations are presented as tables. The bases that are within range of the target are visible by zooming into the map. In addition, the resources available at these bases are presented in tables. (See Figure 2.2.)

A *Map-Based Query* provides two levels of filtering:

- The first level filters out everything except the data that pertains to the objects currently on the map.
- The second level filters out data (both resources and targets) that would not pertain to missions planned against targets currently displayed on the map.

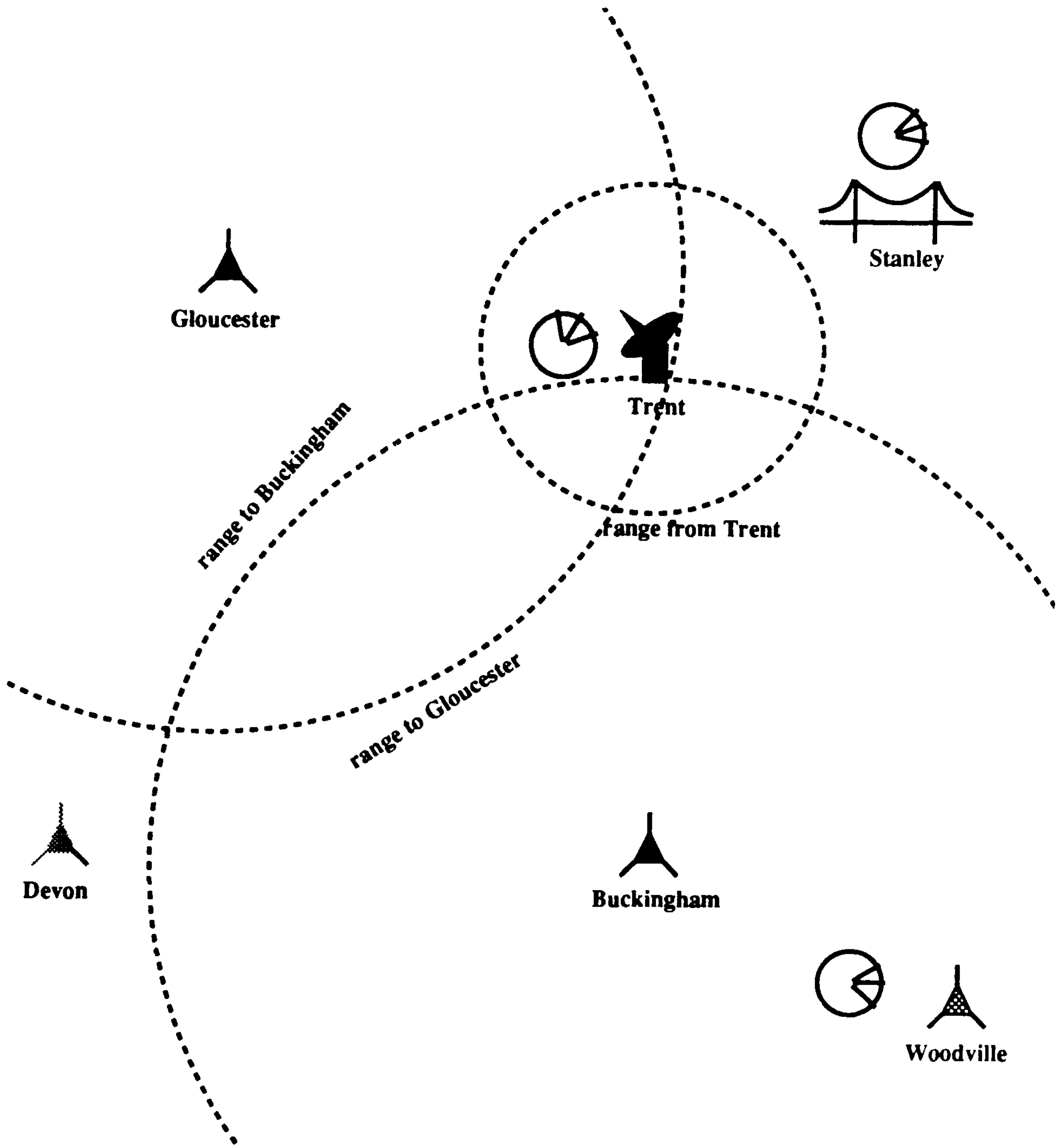
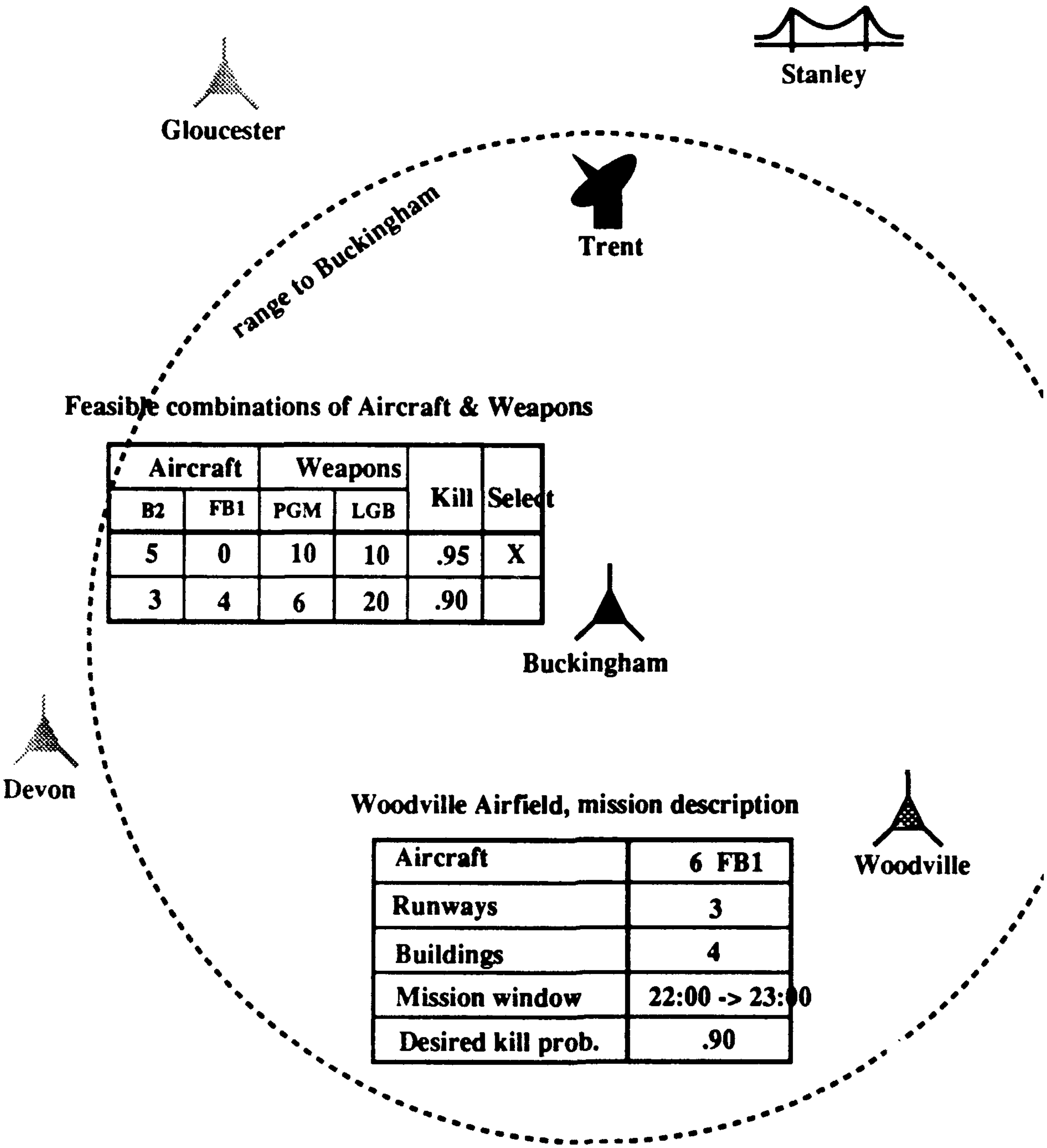


Figure 2.1 Targets, Bases, and Mission Windows



Aircraft		Weapons		Kill	Select
B2	FB1	PGM	LGB		
5	0	10	10	.95	X
3	4	6	20	.90	

Aircraft	6 FB1
Runways	3
Buildings	4
Mission window	22:00 -> 23:00
Desired kill prob.	.90

Figure 2.2 Target Query: Feasible Weapons and Aircraft

This *context sensitive display filtering* prevents the planner from being overloaded with information and having to search through large volumes of data or directly querying the database. In addition, the visual cues from the map in the background provide the planner with the spatial information that is itself the basis for planning.

Once a target is selected, the weapons and aircraft combinations that can be effectively deployed against that target and are available from bases within range are presented to the planner as a table. Each row of the table represents a resource combination (possibly from multiple bases) and an associated kill probability. Using a mouse/cursor, the planner can select a particular resource combination which is to be deployed against a desired target. Additionally, the resource combinations from all currently planned missions are displayed. This information conceptually is a 3–dimensional matrix. It is presented in spreadsheet form with each cell in the spreadsheet corresponding to a single mission–resource combination. A row in the feasible weapons and aircraft table corresponds to a cell in this spreadsheet. The third axis of this spreadsheet displays time and is used for scheduling mission execution, as described in Section 3. (See Figure 2.3.)

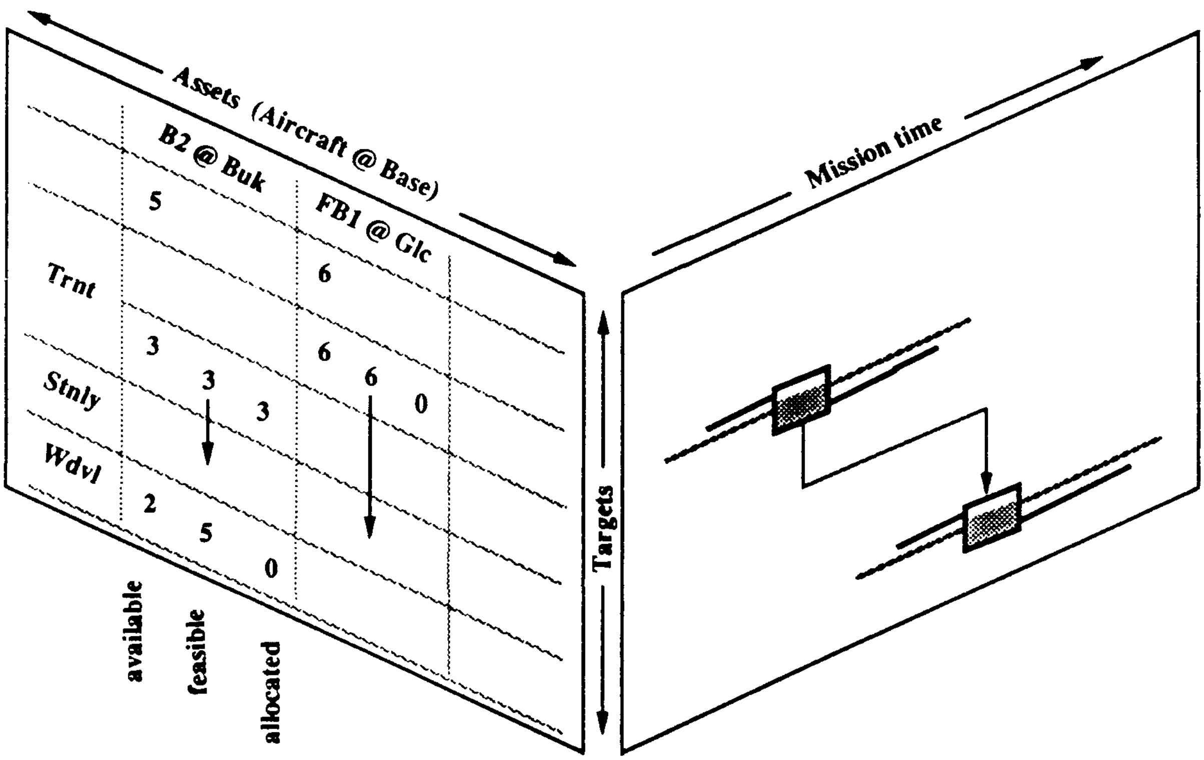


Figure 2.3 Resource Allocation and Mission Scheduling

3 Mission Scheduling

When scheduling missions, planners must keep track of many interrelated goals and constraints. A pool of aircraft exists at each air base from which must be allocated one or more aircraft to attack one or more targets. Constraints between different missions must be accounted for and overall resource allocation must be displayed, so resources are not overcommitted or underutilized. Sensitivity to schedule slippage should be apparent from the display.

For the purposes of this discussion, a *mission* is a collection of related *sorties*. Each sortie consists of one or more of the same type of aircraft, flying from the same air base, attacking the same target. In our examples, sorties are the fundamental unit of discussion.

Each sortie has three types of properties that reflect how it is displayed or how it interacts with other sorties. It has *geographic properties*, such as what base it is leaving from, what target it will attack, and where it will interact with other sorties. It has *temporal properties*, such as how long it will take to fly the sortie (when its aircraft will be unavailable) and when it will interact with other sorties. Finally, it has *resource properties*, such as what impact it will have on various resource pools (fuel, aircraft, or weaponry of various types, for instance). The planner must have all this information available to effectively schedule missions.

In this section, we present three interactive objects that aid planners in interactive mission scheduling: the tot slider, the constraint, and the resource plot.

3.1 Representing Sorties

The most fundamental information about a sortie is its duration. We have developed a graphical icon, called a *tot slider*, similar visually to a horizontal scroll bar, that represents the time information about a sortie. It displays the *mission window* (the acceptable time window during which aircraft can be over the target), the current planned *time–over–target*, the outgoing and returning *flight times*, and the *turn–around time* (the time it takes to refuel and rearm the planes).

Referring to Figure 3.1, the mission window is the solid blue horizontal line, the time–over–target is represented by the green and red box, the outgoing and returning flight times are the blue horizontal dashed lines attached, respectively, to the northwest and southeast corners of the box, and the turn-around-time is the horizontal green dashed line that follows the return flight time. The sensitivity of the sortie to schedule slippage is indicated by the relative amount of the box that is colored red. As the planned time–over-target approaches the end of the mis-

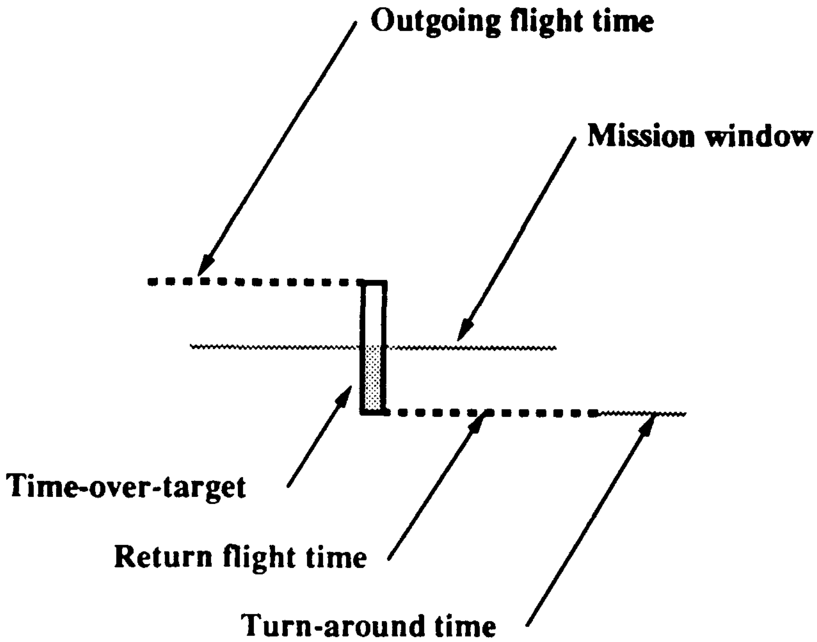


Figure 3.1 Example TOT Slider

sion window, more of it becomes red. If the box is moved completely off one end of the mission window or another, the box, the flight times, and the turn-around-time are all colored red. The relationship between the flight times, the time-over-target, and the turn-around time is fixed. Those objects can be moved horizontally, as a unit, along the mission window.

Describing a sortie in the OSCAR environment is easy. The following LYMB statement defines and initializes a tot slider named "t1".

```
tot_slider new: t1
      label = "Trent Radar"           -- Visible label
      row = 20                        -- Vertical position
      units = 2                       -- How many planes?
      tot = 200;                      --     What     is     the
time-over-target?
```

It will have defaults for the time and duration of the mission window, as well as the flight times and time-over-target.

3.2 Tactical Constraints

Many missions consist of more than one sortie. Coordination is required so that intermediate targets are attacked in proper sequence and that rendezvous take place properly. We have developed an object that defines a constraint between two tot sliders. A constraint associates two tot sliders, the predecessor and the successor, and defines a time difference that some part of predecessor must occur before some part of the successor. If the constraint is violated, an alert action is executed, which normally displays the constraint. (Usually, when a constraint is satisfied, it is not drawn.) A constraint between two tot sliders, t1 and t2, is shown in Figure 3.2. The length of the horizontal segment represents the length of the constraint.

Definition of a constraint between two tot sliders, t1 and t2, using the LYMB script language, is shown below:

```
tot_constraint new: t1_t2
      pred = t1                       -- predecessor
      succ = t2                       -- successor
      pred_msg = "tot?"               -- need t1's time-over-target
      succ_msg = "tot?"               -- ditto for t2
      delta = 30                      --   req'd   minimum   time
difference
      alert_action = alert;           -- constraint violation action
```

If you want the constraint drawn on the screen when it is violated, the simple alert action below will suffice:

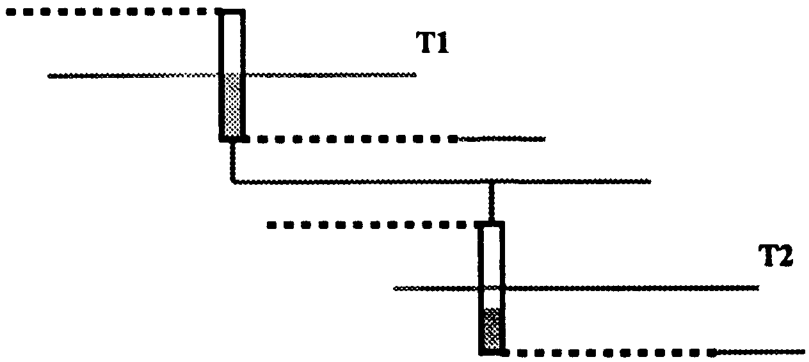


Figure 3.2 Two TOT Sliders Linked by a Time Constraint

```
actions new: alert
      tick_actions = "t1_t2 draw!;"; -- draw the constraint as the
only action
```

More complicated actions can be conceived, however, such as forcing the constraint to be satisfied by moving one tot slider or the other as part of the alert action.

The tot sliders t1 and t2 must know about the constraint object, so they can notify it when they have changed:

```
t1 members = (t1_t2);
t2 members = (t1_t2);
```

3.4 Resource Availability

Nobody has infinite resources at their disposal. In order to plan missions effectively, a balance must be struck between the limited resources available and the targets to be attacked. Planners must know the type and numbers of aircraft at each base so they can ensure that no resources are overcommitted and that the planned schedule is not overly sensitive to slippage.

By themselves, tot sliders do not identify overall resource utilization. Each tot slider is a graphical representation of a single sortie and is used primarily to present timing information. A graphical representation of resource levels as a function of time is necessary.

We developed a simple two-dimensional plot that can be used to represent the allocation of aircraft or other mobile resources. Through LYMB, these resource plots are associated with a number of tot sliders. As a tot slider is moved to modify mission times, the resource plot associated with that tot slider is redrawn to reflect the change in allocation. Normally the plot line is drawn in black, but when a resource is overcommitted, the segment of the graph that is negative is drawn in red. See Figure 3.3 and the video for more details.

In the same way that a constraint is associated with two tot sliders, a resource plot can be associated with multiple tot sliders. Definition of a resource plot that displays the usage of F-16s is shown below:

```
tot_resource new: f16s
      units = 50                        -- how many?
      row = 100                         -- vertical position
      members = (t1,t2,t3)              -- what sorties use F-16s?
      label = "F-16 Usage";             -- visible label
```

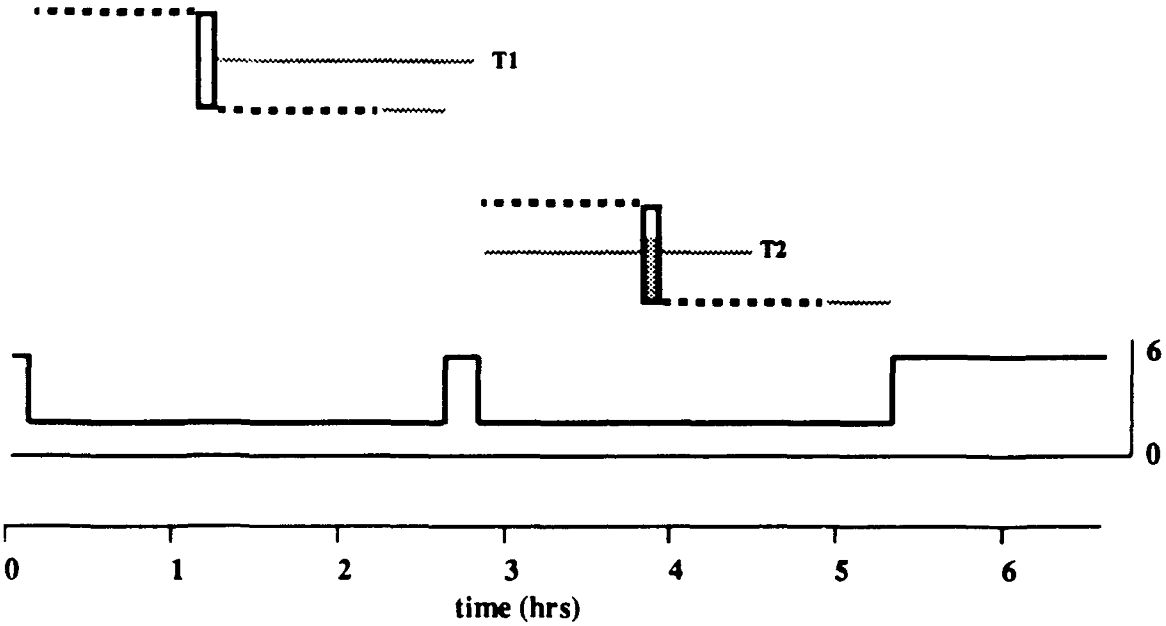


Figure 3.3 Resource Plot Depicting Resource Allocation

The tot sliders must also know about the resource plot, so that when they change, they can notify the resource plot to redraw itself:

```
t1 members+(f16s);
t2 members+(f16s);
t3 members+(f16s);
```

4 Resource Allocation Networks

A **resource allocation task** for Air Force planning may be specified as a collection of *targets* and *bases* geographically distributed over some *gaming area*. A database specifies the *attributes* and the *strategic value* of each target, the alternative *combinations of resources* (such as aircraft and weapons) which can be effectively used against each target, and the *probability of success* for each resource combination which may be used against a given target. The database also specifies the *availability* of each type of resource at every base at the start of the planning period. We are required to make a feasible allocation of the given resources to the given targets and to *maximize the risk weighted return over the entire set of missions*.

We have developed **Resource Allocation Networks** for visualizing the resource allocation alternatives for a collection of missions with competing resource needs and for representing the interdependence between those missions. Our technique permits the user to visualize the probabilities of success of the feasible alternative plans and to use the pre-assigned strategic values of the given targets in order to obtain a resource allocation plan which provides the best risk weighted return over the entire set of missions.

4.1 Traditional Techniques

Job scheduling and *resource allocation* are among the most important concerns of planners and factory managers. Traditional tools for these tasks include **GANTT** charts and **PERT** charts [3, 4]. The combined use of these techniques permits the visualization of: (i) the variation in the usage and availability of several resources spanning the duration of the planning horizon, (ii) the sequential dependence amongst related tasks, and (iii) the reuse of resources across consecutive tasks. There is also some conceptual similarity between our technique and **Petri Nets** [5], although the latter have not been used in the context of resource allocation tasks.

The main novelty of our technique is that it permits a better analysis and visualization of the *risks and rewards* associated with each of the several feasible resource allocation alternatives. The following specific features of our technique are designed to support risk-reward analysis during resource allocation and planning:

- A network of resource and target icons are interconnected in order to visually indicate the current allocation of resources to specific targets, as well as all as to show all the feasible alternative uses for each resource (including those which are currently infeasible).
- Iconic images are used to indicate the type of each target.

- Meters indicate the strategic value of each target.
- Meters indicate the probability of mission success which is attainable by using each of the alternative combinations of resources which can be possibly deployed against a given target.
- Meters indicate the costs and rewards associated with the *currently allocated deployment* of each resource, as well as all of its ***potential uses***.

4.2 Defining Resource Allocation Networks

Figures 4.1 and 4.2 show an example of a resource allocation network and the meaning of the symbols. On the network, time flows from left to right, as in a PERT chart. Resources are represented by green octagonal icons and show the resource type, location and available quantity. Stacked boxes beside a resource indicate simultaneous feasible (and infeasible) allocations of units of that resource to different targets.

A target is represented by a red rectangular icon and shows its location and type. Its strategic value is indicated by a green vertical meter to its right. Stacked boxes beside a target indicate alternative weapons and aircraft combinations which may be used against that target. A box with a green dot indicates the currently selected plan for that target. An orange horizontal meter next to the target indicates the kill probability for that target using the currently selected plan. Pop-up meters can display the kill probabilities for alternative (unselected) plans for that target.

A blue line from a resource box to a target box indicates the number of units of that resource which are allocated to that target. Since time is represented along the horizontal axis of the network, the horizontal projection of the line from the resource to the target indicates the duration of the mission. A resource icon can be indexed with a time value and replicated to the right of its allocated target, in order to indicate the time at which the resource becomes available for reuse.

4.3 Managing Conflicting Resource Needs

A *resource allocation network* can be used for the following tasks:

1. To view feasible alternative weapon and aircraft (W&A) combinations which can be used against a desired target, as well as the kill probabilities associated with each option.

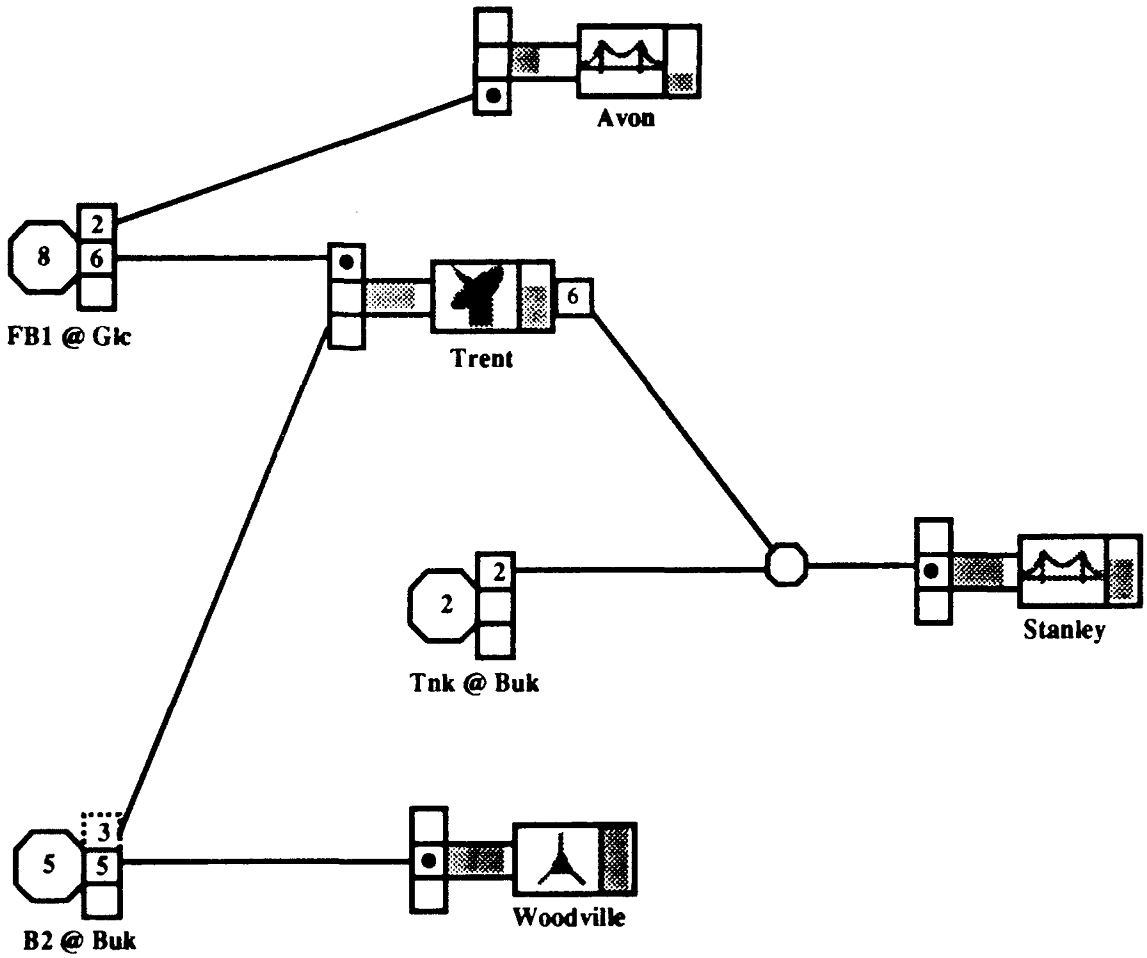


Figure 4.1 Resource Allocation Network

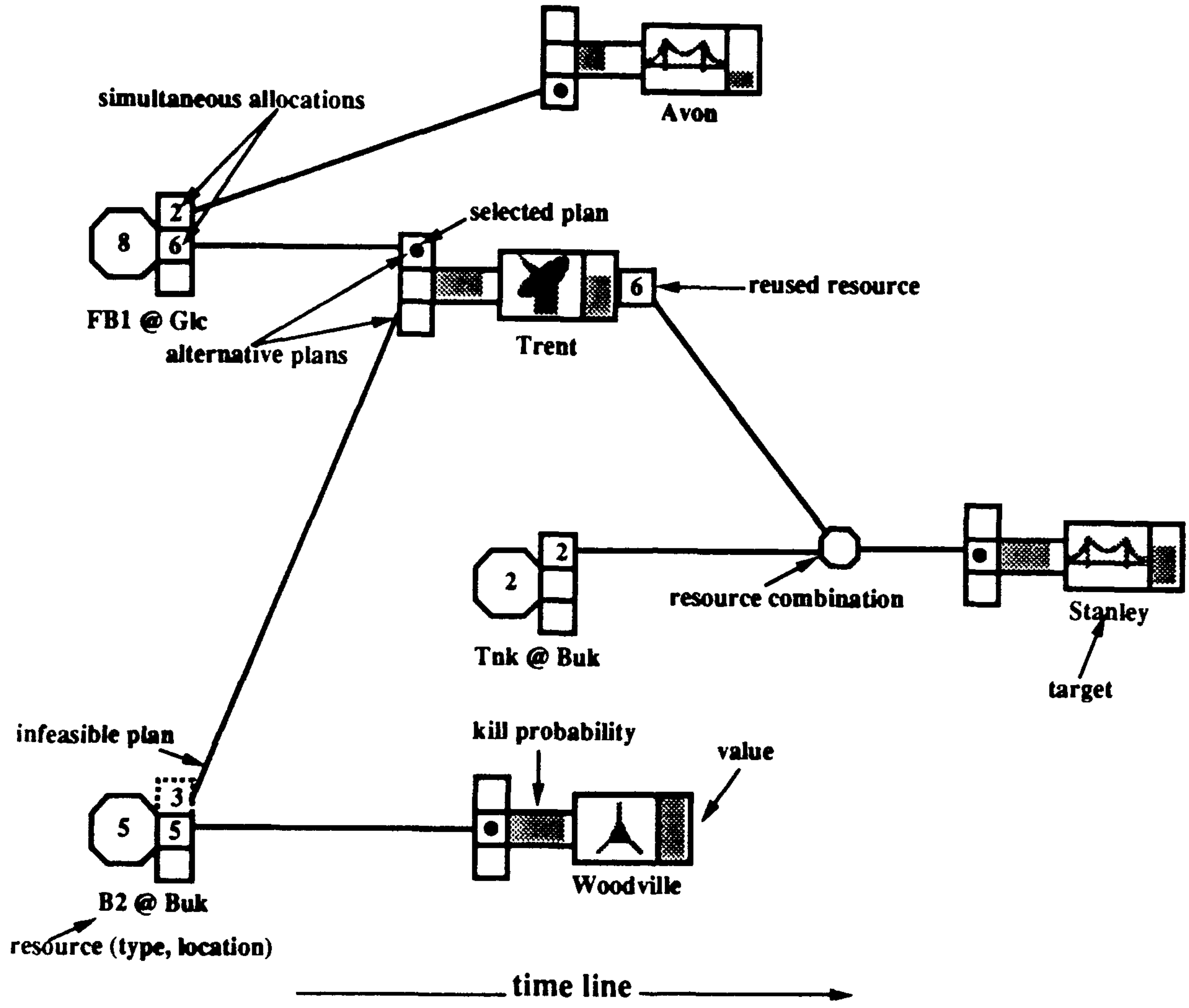


Figure 4.2 Symbology for Resource Allocation Network

The input ports of a target box define all the weapon and aircraft alternatives which are technically feasible. The edges coming into the input ports of a target can be traced backwards to determine whether the desired W&A option is feasible in terms of the current deployments of W&A to various missions.

2. By tracing through successive edges in the network and modifying their end points, it is possible to transform an infeasible allocation into a feasible allocation (if one exists). If all the W&A options for a target *A* are infeasible on account of the current deployments of resources, then we select one (technically feasible) input resource port *R* of *A* and determine another mission *B* such that *R* is deployed for *B*. We then determine if any other feasible W&A option can be deployed for *B*, deallocate *R* from *B*, and deploy *R* for *A*. Thus we see that redeploying resources in order to make more missions feasible simply involves tracing through *paths* (consisting of connected sequences of nodes and edges) in the resource allocation network.

3. To view alternative deployments of a selected resource and the risks and rewards associated with those options.

4. To view dependencies between missions which are dictated by the reuse of resources. Additionally, a tactical dependency between a pair (*A*, *B*) of missions can be visualized by connecting a dummy resource after the end of mission *A* and connecting the dummy resource to the mission *B*, which is the successor of *A* in the given tactical dependency.

4.4 Risk Reward Analysis

The output ports of a resource box are connected to all the targets against which it is technically feasible to deploy that resource. For a given deployment, there is a cost, which is the sum of the cost of using that resource, and the risk weighted cost of losing that resource. The value which results from deploying a particular resource is the product of the kill probability for that mission and the intrinsic strategic value of the target of that mission. The rewards for a mission M can be augmented by the values arising from missions which depend on the success of M. By examining the pop-up risk–reward meters associated with each output port of a resource, it is easy to determine whether a given resource is being used to the best advantage, consistent with the given constraints of the planning task. An example of such a query is shown in Figure 4.3.

4.5 Impact of Weather on Planned Missions

The pop-up risk reward meters (RRM) associated with each mission can be augmented with additional values to account for the effects of unfavorable weather. For example, a given mission *M* may have a predicted kill probability *p* if the weather is clear. Further, the kill proba-

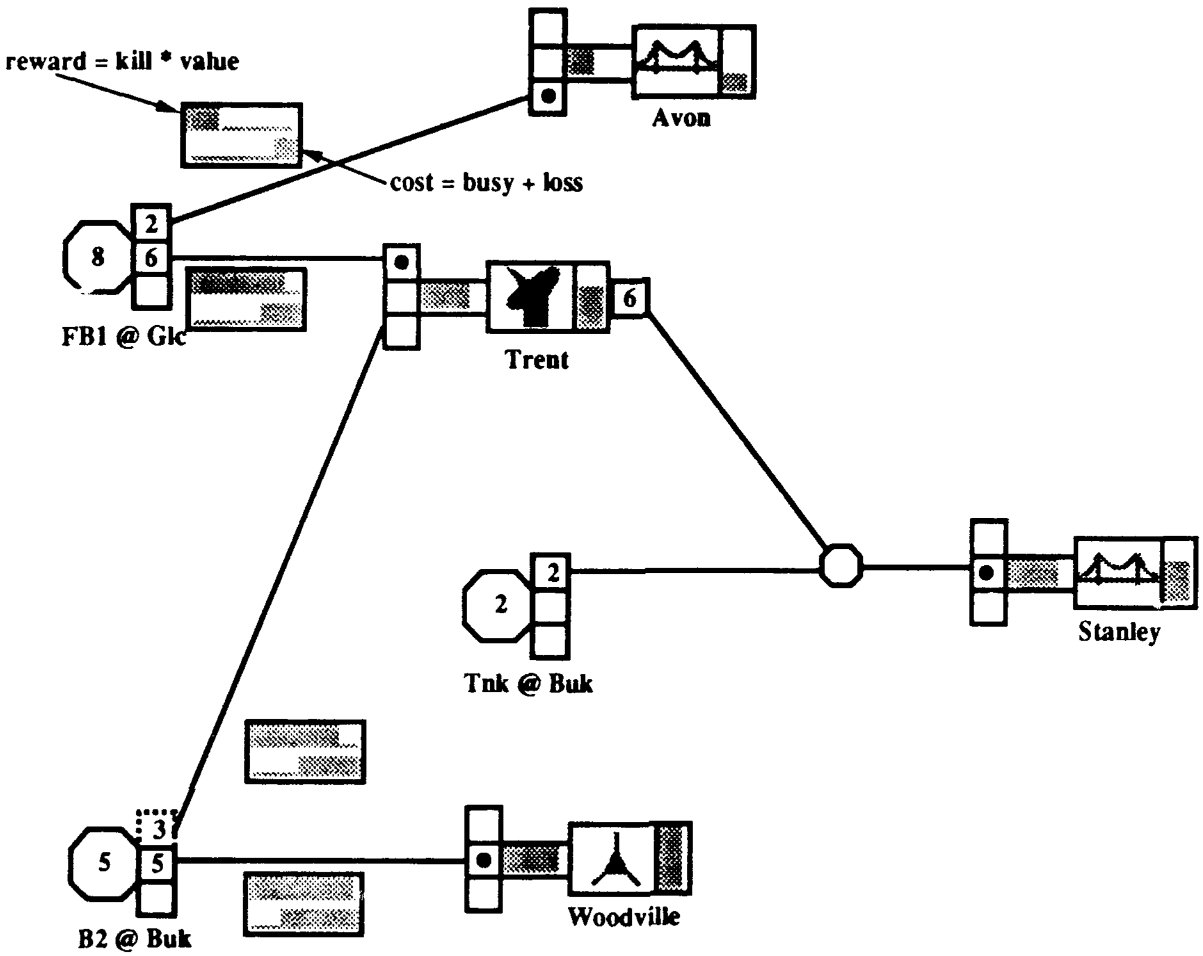


Figure 4.3 Analyzing Cost Benefit & Risk Tradeoffs

bility may drop to q if it rains, and drop to r if it snows. The RRM for M would now show the three reward values for the kill probabilities p, q and r.

The user can interactively sketch out the extent, path, and timing for a predicted weather phenomenon such as a snowstorm. The extent and path can be specified in an obvious manner as an overlay on the map of the gaming area. The timings can be specified by using a clock face,or by means of a pop-up dialog box, located at the start of the trajectory of the weather phenomenon, as well as at desired intermediate points along the path.

The system can simulate / animate the progress of the predicted weather phenomena (such as snowstorms or rain). It can also simulate the progress of the planned missions. The simultaneous weather and mission simulations can be used to determine which missions are unfavorably impacted by the predicted weather phenomena. This results in an updated display of the risks and rewards associated with each mission represented by the resource allocation network.

5 Coordinating Missions

A mission is typically specified at the Force level (above the unit level) by the end points of its trajectory and its start and end times. Also given are a set of *waypoints* along the trajectory of each mission, which are typically used for coordinating with other missions or offensive / defensive activities. Additional constraints on individual missions are defined by specifying nearness / minimum separation with respect to friendly / hostile positions, by topographic constraints placed on feasible flying routes, and by the performance constraints of the aircraft. We are required to generate a set of coordinated mission plans (including high level flying routes and times), such that the missions individually and collectively satisfy the given constraints.

We have developed an interactive graphical technique for visualizing and modifying *spatial and temporal coordination constraints* among a set of inter-dependent missions. Using our technique, the user can interactively sketch the spatial trajectories of the desired missions on a displayed map of the gaming area, and also define desired constraints in the *locations and timings* of a subset of the events which constitute the given missions.

A mission can be viewed abstractly as a sequence of events, each of which is specified by its location and time. The constraints specify how near or how far apart two events can be. The pair of events thus constrained may be chosen from the same mission or from two distinct missions. Further, the specified constraint may be the geographic distance between the two events or their separation in time.

The central problem is the ability to visualize a sequence of *events*, each of which is specified simultaneously by its *location and time of occurrence*. Given several *missions*, each of which is a sequence of events, we are required to relate two events from the same mission or from different missions. The novelty of our technique is the ability to visualize and interactively satisfy the spatial and temporal constraints on the given set of missions.

5.1 Constraint-Based Mission Planning

The object of our technique is to enable planners: (i) to visualize the spatial and temporal coordination constraints present among a given set of inter-dependent missions and (ii) to interactively generate a set of coordinated mission plans which individually and collectively satisfy the given constraints

The user is presented with a map of the gaming area on the screen, containing labelled icons representing the targets, bases, and other natural and cultural features of interest. The user can interactively sketch out the locations or trajectories of planned events / missions. For a multi-event mission, intermediate events of interest can also be identified with labels. Events at distinct locations can be identified as occurring at the same time by linking a pair of such events with a line, or enclosing a set of events with a closed line.

Figure 5.1 shows an example of the usage of our technique for coordinating a refuelling mission with a two step bombing mission. A fighter plane starts out from the base at Gloucester (1) along the upper trajectory, bombs the radar site at Trent (2), is refuelled en route (3), and then bombs the bridge at Stanley (4). A tanker takes off from the base at Buckingham (A) along the lower trajectory, passes through location (B), refuels the fighter en route (C), and returns to its base. The dotted blue lines identify the pairs {(2, B), (3, C)} of events which are pair-wise simultaneous. The two blue lines at the bottom show the time lines of each mission, with time increasing to the right. The peaks in the time lines correspond to labelled events, and the width of each peak defines the permissible slack in the time of occurrence of the event.

The location of an event need not be a point, but could also be defined as an enclosed region of the map. Additional spatial or temporal inequality constraints between selected pairs of events may be defined interactively by the user. Some constraints on the locations of events or flight paths may be predefined by means of generic constraints relating to the underlying terrain or cultural features or the current status of the *line of control*. Similarly, a time constraint between any two locations of a flight path can be automatically introduced by the software, in response to the performance constraints of the aircraft.

5.2 Using a Constraint Based Planner

Once the constraints have been defined, an underlying constraint solver attempts to solve for the locations and time of occurrence of each event based upon the given initial values and the given set of constraints. Violated constraints are highlighted and brought to the user's attention. At this point the user can interactively modify the location and/or time of occurrence of selected events and see the impact on the satisfiability of the complete set of constraints. As an example, a selected flight segment may require the aircraft to fly faster than its maximum velocity, and this would result in that flight segment being highlighted (possibly by changing its color). Then the user could drag either end of a constraint line relating to either end of that segment, and thus cause a change in the defined time or location of one of the offending events. After a sequence of visual iterations, the user can derive a feasible set of missions which satisfies all the given constraints. After the mission planning stage, animation can be used to visualize the progress of the planned missions over simulated time.

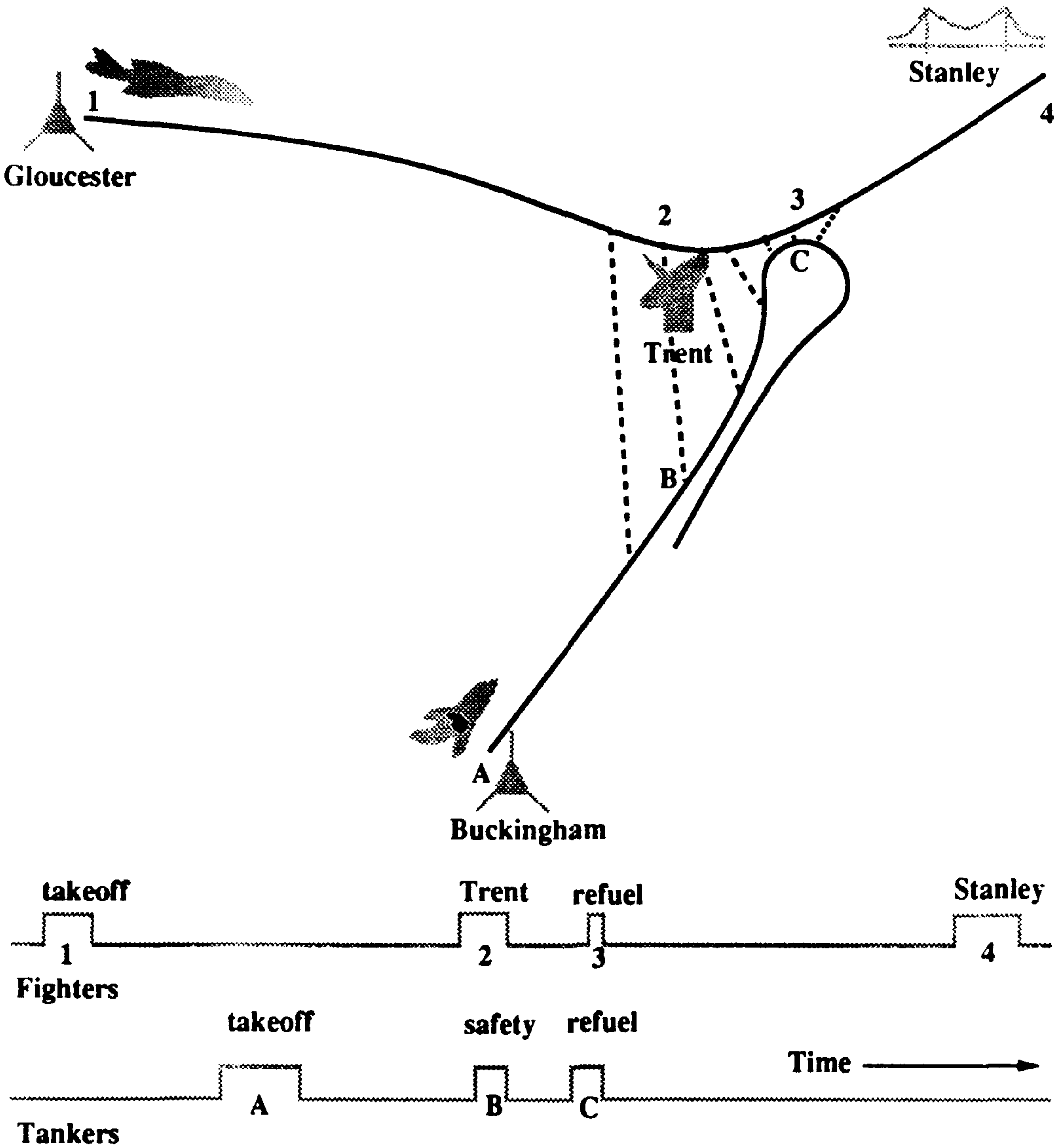


Figure 5.1 Coordinating Refueling with Two-Step Bombing

We have devised a mocked up mission planning scenario for force level planning and devised a graphical prototype animation of the execution of a set of planned missions. Figure 5.2 shows a frame of the animation of the missions planned in Figure 5.1. The frame shown corresponds to the time at which the fighter and the tanker rendezvous for refuelling after the bombing of the radar at Trent. (Notice that the radar is shown dotted.)

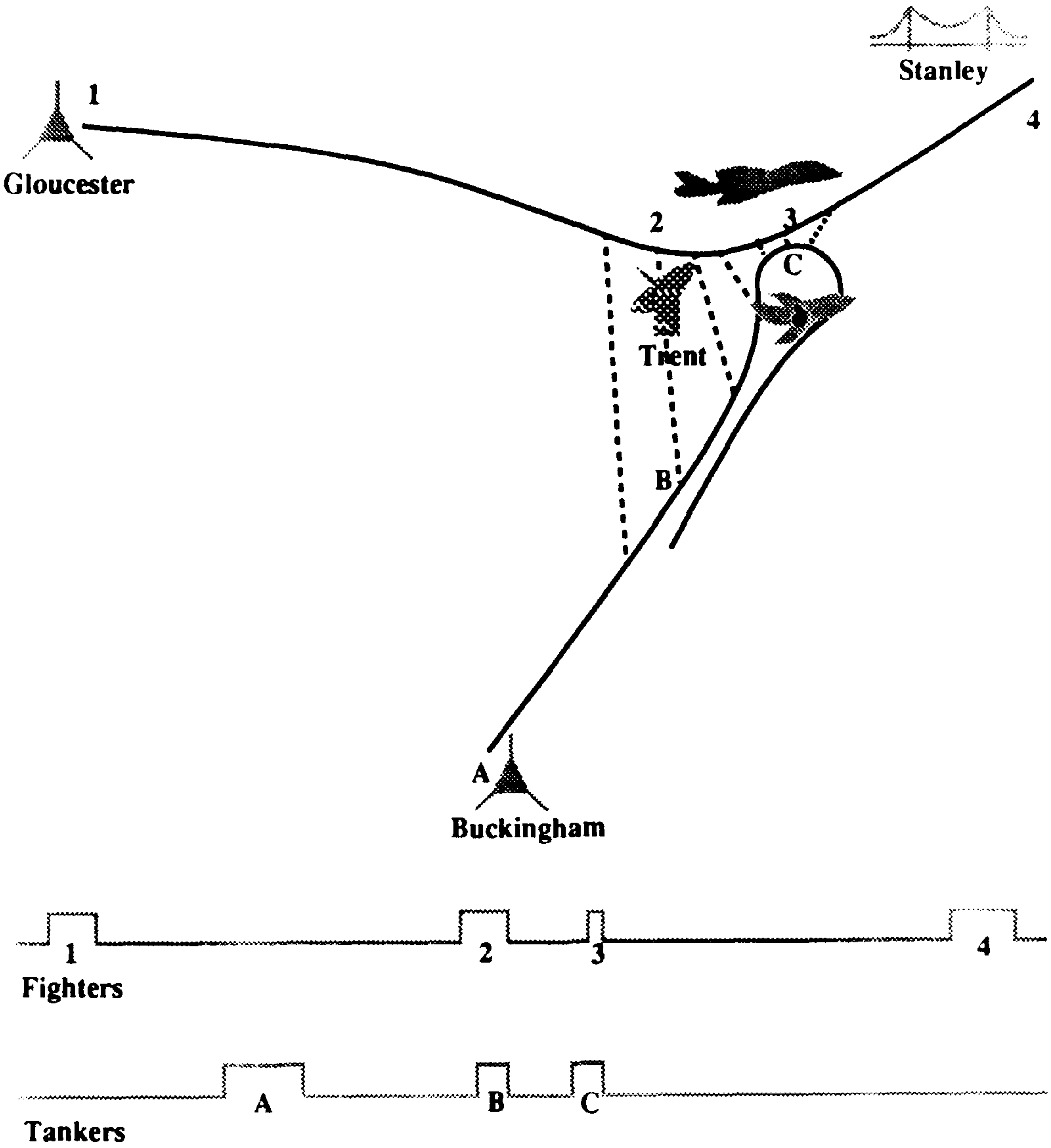


Figure 5.2 Mission Coordination Animation

6 Strategic Situation Assessment

Assessing the strategic value of a set of mission plans in a theater is difficult. Many factors interact to contribute to the overall strategic state of the area. Timing of missions and the types of aircraft used, deployment of mobile air- and land-based resources such as AWACS or ground-based radar and missiles, and levels of weaponry and fuel reserves, all contribute to the overall "value" of the situation. Unfortunately, even if a single set of equations could be developed that took into account all the various factors, the result would not be a single number. Because a theater of operations is a large geographic area, the representation of the value over that area is necessarily multiple-valued. To understand this complex multivariate function, it should be displayed graphically in a time-dependent manner. The resulting display is at least four-dimensional (x and y dimensions over the region being evaluated, one or more values sampled discretely over the region, and changes to the value(s) as a function of time).

If the value at a location (say, the defensive value of a radar site) has an effect over a non-zero subarea of the gaming area, then we say that information is *continuous*. If the value at a location has no effect over a non-zero subarea of the gaming are, then we say that information is *discrete*. Some data can be viewed as both discrete and continuous. (We have yet to come up with any information that is purely discrete.) Fuel supply at an air base is a discrete quantity from the viewpoint of a supply officer trying to determine where to replenish fuel supplies, but from a planner's viewpoint it can be viewed as continuous, since the fuel supply affects the range of defensive and offensive capability in space and/or time.

Military planners might ask a number of questions about discrete or continuous data. Some representative questions are:

- Given some useful function that can be evaluated or estimated at a discrete set of points in a region, such as the defensive value of a single air base, how can the overall defensive value of a larger region be understood?
- How does the defensive capability change over time, as planes depart or return or other resources are consumed or replenished?
- If mobile resources (such as truck-mounted radar or anti-aircraft guns) are moved, will the defensive capability of the regions they leave be compromised? If so, how might other resources be redeployed to fill the gap?
- Given a discrete function, such as fuel reserves at air bases in the theater, how can they be viewed as a function of time?

- Given current plans, will fuel reserves become critical at any air bases over the next two days?

Given a set of application-specific resources, with a known geographic and temporal variation in their availability, and a way to combine the tactical and strategic value of those spatially distributed resources, we can view the resulting values as three-dimensional surfaces that vary with time. We call these surfaces *sitmaps*, short for situation maps. We believe sitmaps can be used to depict several multi-valued strategic and logistic functions, such as offensive and defensive capabilities or weakness of friendly or enemy forces, or fuel or weapon reserves at air bases.

Each resource has an offensive and defensive capability that varies with distance from its current location. For instance, a radar can only "see" a certain distance, and the probability of detecting a target falls off with increasing distance from the radar. Over the area scanned by the radar, some defensive value can be computed. A radar has no direct offensive capability. Similarly, an aircraft has both offensive and defensive capabilities that are a complex function of its position, fuel and weapon load, and whether it is flying on a mission. An air base may have defensive and offensive capabilities that are some complex weighted sum of the defensive and offensive capabilities of its resources. An aircraft that is flying an offensive mission may contribute little or nothing to the defensive capabilities of its home air base.

Multiple ways to combine simpler resource values into more complex ones are needed. Typically, this will be an application-specific process. The display system described in this section is independent of the way in which resource values are computed. The only inputs are a resource's value, its location, its radius of effectiveness (at what distance from the resources its value decays to zero), and how its value decays as you move away from it. Similarly, combining all the resource values into a set of geographically dispersed values is an application-specific task.

Figure 6.1 shows an example planning display using tot sliders to depict five sorties originating from five different air bases. The display consists of a number of tot sliders and a *time selection slider* (the black box at the bottom of the display). Each tot slider defines the time during which a set of aircraft will be flying a mission and will not contribute to their air base's defensive capabilities. In the figures, we are therefore estimating the defensive capability over a geographic region based upon the defensive capabilities at a discrete set of air bases. The time selection slider is used to indicate the point in time for which the sitmap is computed and displayed.

Figure 6.2 shows two sitmap representations, one displayed as a three-dimensional surface, the other as a two-dimensional contour plot, that reflect the state shown in Figure 6.1, at

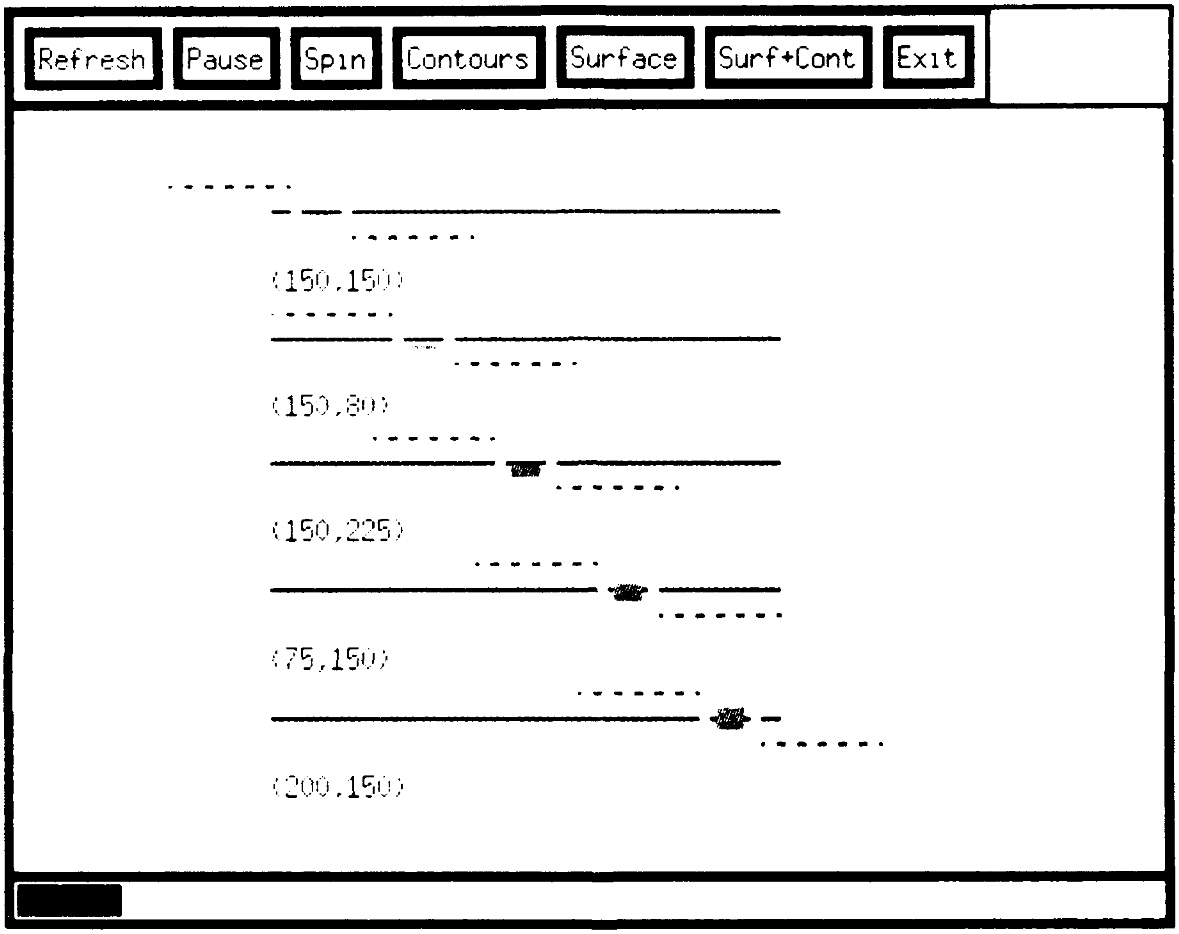


Figure 6.1 User Interface to Sitmap Manipulation

[illegible] Configuration in Figure 6.1

the time represented by the time selection slider. A contour plot may be preferred over a three-dimensional surface when

- detection of extremely subtle changes in contour values is necessary,
- less powerful graphics display hardware is being used, or
- the display of actual map data is also desired.

As the user moves a slider, the sitmap is updated to reflect the projected resource values at the time indicated by the time selection slider. The effect of time selection slider movement is shown in Figure 6.3. It has been moved to sit just beneath the box of the tot slider labelled *(200,150)* in Figure 6.1. The effects of changes in resource allocation are shown in Figure 6.4. The time selection slider is in the same position as Figure 6.3, but the fourth tot slider (labelled *(75,150)* in Figure 6.1) has been moved to the left edge of the mission window.The video segment shows an example of interactive control of the sitmap described in Figures 6.1 through 6.4.

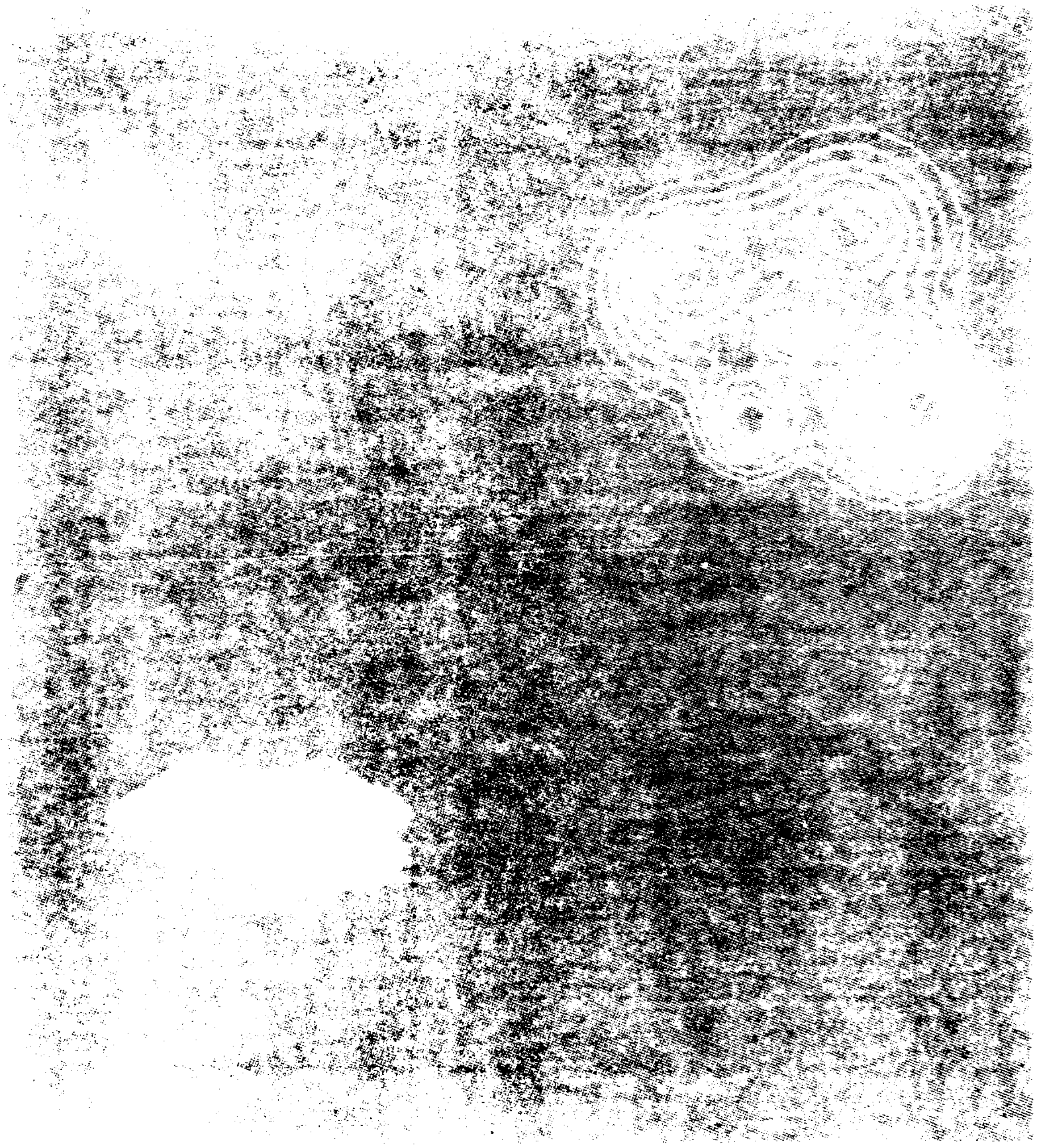
6.1 Basic Techniques

To compute the sitmap for an area, values on a regular grid that covers the area of interest are interpolated from the known values at discrete points in the region. We have used two different interpolation functions, one based on area coordinates that relies on a triangulation of the points, the other a distance-weighted sum of the known values. We prefer the distance-weighted sums because they take into account all known values, not just those at the vertices of the triangle enclosing the query point. Other interpolation schemes, such as averaging or minimum or maximum functions can be used as well.

How the value approaches zero is governed by a function of the distance from X,Y. To compute the value at any point *x,y* in the region, we simply compute a distance-weighted sum of the values of all objects in the region. In pseudocode this looks like

```
for all x in the region
      for all y in the region
           v(x,y) = 0
           for all p in the set of objects in the region
                 d = distance from (x,y) to the current object
                 v(x,y) = v(x,y) + V(p) * (1 - d/R(p))^E
```

V(p) is the value at point *p*. *R(p)* is the radius of effectiveness of *p*. *E* is an exponent that governs how the expression *(1 - d/R(p))* approaches zero. For example, radar sites would have an exponent of 2, since they obey an inverse square law.



Time Slider

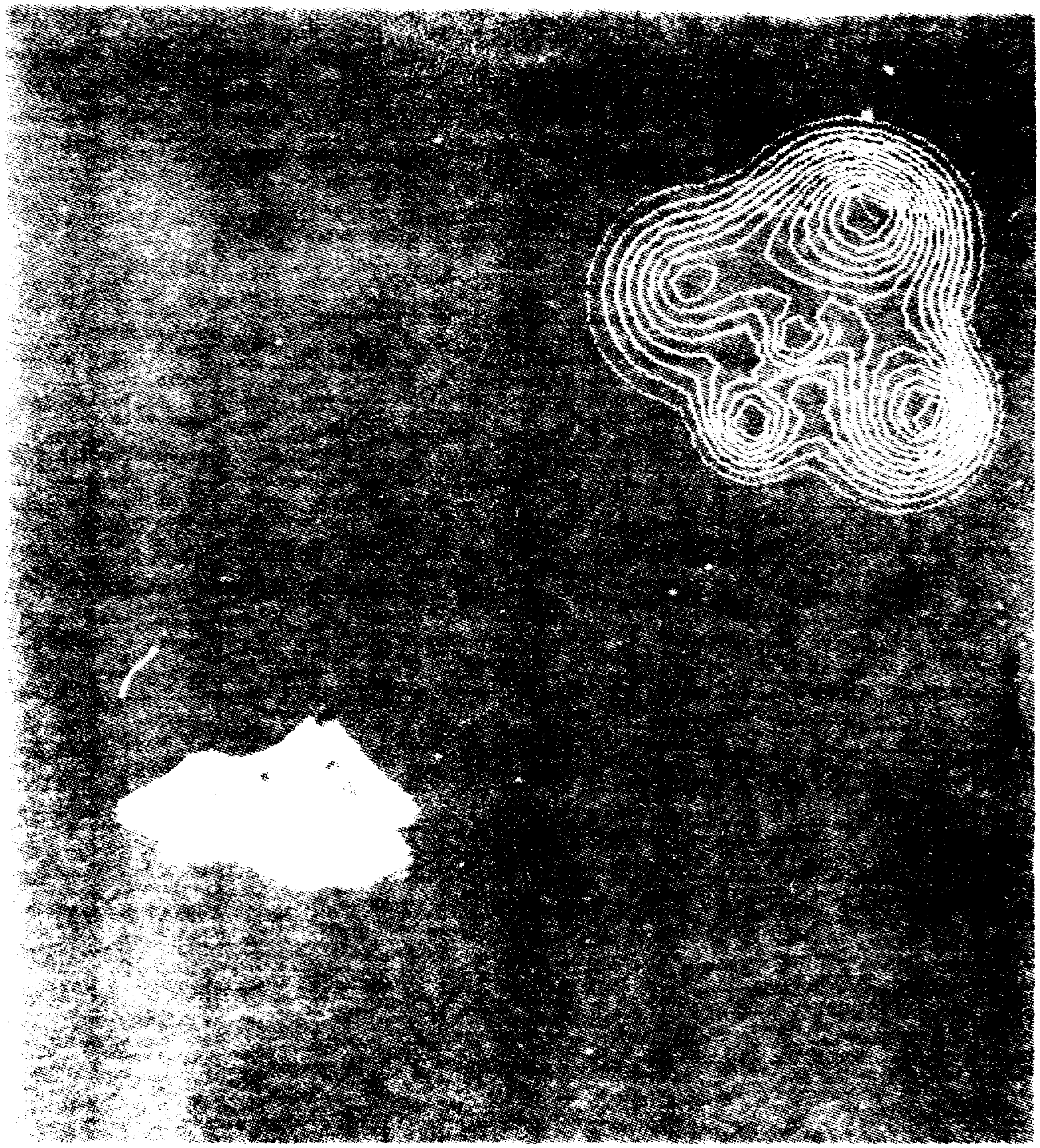


Figure 6.4 Sitmaps After Moving Fourth TOT Slider

Once the values on a regular grid of x,y points in the region have been computed, they are displayed as a surface in three dimensions, with the value at x,y taken as the z coordinate. Currently, we color the surface based upon the elevation data as well, but a second value could just as easily be associated with color.

To show the effects of time or changes in mission plans, the values at the known points are reevaluated and the surface is recomputed and redisplayed.

6.2 Multi-Valued Sitmaps

The current sitmap implementation maps both the z coordinate and surface color to a single "value". Future animations will map color to other values, allowing multiple functions to be displayed. Other display techniques common to scientific visualization, such as hedgehogs (for displaying vector-valued functions) or streamlines, may be used in the final animation.

6.3 Animated Sitmaps

Besides changing the time at which resource changes occur or the simulated time at which the sitmap is computed, the resources themselves may also move. For instance, a mobile radar unit may move, yielding a different set of defensive capabilities, or tankers may travel along a known trajectory to a rendezvous point, increasing the reach of offensive aircraft and thus changing the sitmap.

Viewing changes in a sitmap as resources move or simulated mission time progresses is best done using animation. The example in the video computes the static sitmap as a function of the user's inputs, but the final video will demonstrate animation of the sitmap through time.

7 Conclusions

In this report we have described several novel graphical techniques which can assist an Air Force planner to visualize the dependencies among the set of planning tasks and also to interactively derive a set of missions which are consistent with the predefined goals. We list below the specific steps of the planning process where our interactive graphical techniques can provide support to the planner.

- The user limits the data to be used by zooming and panning across a displayed map of the gaming area, and he can select and query targets, bases, weapons and aircraft by pointing and clicking with a mouse at displayed icons.
- Next the user can schedule missions by using *tot_sliders*. The related techniques also enable the user to specify and visualize tactical precedence constraints among several missions and to determine how sensitive the missions are with respect to slippages in their schedules.
- The planner can visualize all the available alternatives for allocating resources to targets by using *resource allocation networks*. This technique also enables the user to reallocate the available resources among the set of planned missions, in order to maximize the number of feasible missions.
- The planner can also perform a cost-benefit analysis of the planned missions by using the pop-up *risk reward meters* associated with the resource allocation networks. In this manner, he can determine whether the resources are being deployed for a maximum benefit or whether some resources can be redeployed against alternative targets in order to maximize their strategic utility.
- If several missions are dependent on each other, the planner can coordinate their plans interactively by using the *coordinated mission maps*. This technique permits the explicit representation, visualization, and manipulation of the geographic and temporal constraints present among a set of interdependent missions.
- Finally, the user can assess the strategic and tactical impact of the planned missions on the geographic and temporal distribution of offensive and defensive capabilities by using the three dimensional *sitmaps*. Sitmaps can also be animated, in order to display the impact of the execution of the planned missions or of planned troop and weapon movements, over the entire duration of the planning horizon.

In summary, we have devised several novel interactive graphical techniques which can assist an Air Force planner during every facet of the planning process. The most fundamental aspect of force level planning is that it is an *iterative process*, during each cycle of which it is important to be able to visualize the inter–dependence among the decision variables, and to be able to gauge the impact of modifying specific decisions. Traditional text based techniques deny the user the power of the interactive graphical medium for visualizing these dependencies, and for gauging the impact of proposed changes.

8 References

1. Yamrom, B., W. Lorensen, and W. Schroeder. *A Practical Introduction to LYMB*. GE Corporate Research and Development. July 1990.

2. Lorensen, W., B. Yamrom, D. McLachlan, and M. Barry. *LYMB Reference Manual.* GE Corporate Research and Development. October, 1986.

3. R.S.Pressman. *Software Engineering: a Practitioner's Approach*, McGraw Hill, 1987.

4. J.Wiest & F.Levey. *A Management Guide to PERT / CPM*, Prentice Hall, 1977.

5. T. Murata. *Petri Nets: Properties, Analysis and Applications*, Proc. of the IEEE, vol. 77 No. 4, pp. 541-580, Apr. 1989.

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.